

AD-A199 308

AGARD-LS-158

AGARD-LS-158

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

AGARD LECTURE SERIES No.158

Computing Systems Configuration for Highly Integrated Guidance and Control Systems

DTIC
ELECTE
JUL 13 1988
S C H D

NORTH ATLANTIC TREATY ORGANIZATION



DISTRIBUTION AND AVAILABILITY
ON BACK COVER

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution is unlimited

88 7 13 026

AGARD-LS-158

NORTH ATLANTIC TREATY ORGANIZATION
ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT
(ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD)

AGARD Lecture Series No.158
COMPUTING SYSTEMS CONFIGURATION FOR HIGHLY INTEGRATED GUIDANCE
AND CONTROL SYSTEMS

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Guidance and Control Panel of AGARD and the Consultant and Exchange Programme of AGARD presented on 20—21 June 1988 in Bourges, France, on 23—24 June 1988 in Neubiberg, Germany and on 6—7 July 1988 in Seattle, USA.

THE MISSION OF AGARD

According to its Charter, the mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community;
- Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Exchange of scientific and technical information;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors.

Published June 1988

Copyright © AGARD 1988
All Rights Reserved

ISBN 92-835-0464-X



*Printed by Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ*

ABSTRACT

Modern military air vehicles have to comply with sophisticated performance requirements. As a result, full advantage must be taken of the rapid advances in computer hardware/software and future micro-electronics technologies.

New design and development strategies must be implemented in order to obtain the overall performance benefits offered by advanced integrated systems for guidance and control, avionics, weapon delivery and tactical performance management.

In a two-day programme this Lecture Series will address some issues which have demonstrated notable and outstanding advances in the field of computing system design, design tools and techniques, computers, data buses, and architectures. In particular, the second day's programme will show how technological advances have enabled the design of a modern computing system architecture. Future trends and new directions will be subjects for round table discussions.

This Lecture Series, sponsored by the Guidance and Control Panel of AGARD, has been implemented by the Consultant and Exchange Programme of AGARD.

Les aéronefs de combat modernes doivent répondre à des spécifications de performances sophistiquées. Il importe donc, de tirer le meilleur parti de la rapidité des progrès réalisés dans les domaines des matériels et des logiciels d'ordinateurs et des technologies d'avenir de la micro-électronique.

Il y a lieu de tenir compte des nouvelles stratégies d'étude et de réalisation, qui permettent de bénéficier des avantages offerts, en termes de performances générales, par les nouveaux systèmes intégrés de guidage et de pilotage; d'avionique et de tir d'armes, ainsi que par les systèmes de gestion des performances tactiques.

Les deux journées de ce Cycle de conférences sont consacrées à l'examen de certains secteurs où des progrès remarquables et exceptionnels ont été réalisés dans le domaine de la conception des systèmes informatiques, à savoir: les techniques et les aides à la conception; les ordinateurs, les chaînes de données et les architectures. En particulier, les présentations de la deuxième journée concernent les progrès technologiques qui ont permis la réalisation d'une architecture de système informatique moderne. Les tendances et les perspectives d'avenir feront l'objet d'une table ronde.

Ce Cycle de conférences est présenté dans le cadre du programme des consultants et des échanges, sous l'égide du Panel AGARD du Guidage et du Pilotage.



| | |
|--------------------|-------------------------------------|
| Accession For | |
| NTIS GRA&I | <input checked="" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

LIST OF AUTHORS/SPEAKERS

Lecture Series Director: I.P.A. L.Guibert
Chief, Integration Section
STTE
129, rue de la Convention
75731 Paris Cedex 15
France

AUTHORS/SPEAKERS

Dipl. Ing. H.J.Kaul
Flugzeuge und Hubschrauber
MBB GmbH
Abt LKE34
Postfach 80 11 60
8000 Munich 80
Germany

Dipl. Ing. H.Lüers
Flugzeuge und Hubschrauber
MBB GmbH
Abt LKE34
Postfach 80 11 60
8000 Munich 80
Germany

Dr Mark T.Michael
AFWAL-ATT-2
Bldg 620, Area B
WPAFB, OH 45433-6543
USA

Mr David R.Morgan
AFWAL/AAAS-1
Bldg 620, Area B
WPAFB, OH 45433-6543
USA

Mr M.Muenier
E S D
55, Quai Marcel Dassault
BP 301
92214 Saint-Cloud Cedex
France

Mr John C.Ostgaard
Chief Advanced Integration Group
System Integration Branch
AFWAL/AAAS-1
Bldg 620, Area B
WPAFB, OH 45433-6543
USA

Dr R.Uhlhorn
Harris Corporation
Government Aerospace Systems Div.
P.O.Box 94000
Melbourne, FL 32902
USA

CONTENTS

| | Page |
|---|-----------|
| ABSTRACT | iii |
| LIST OF SPEAKERS | iv |
| | Reference |
| OVERVIEW AND INTRODUCTION by L.Guibert | 1E |
| INTRODUCTION ET PRESENTATION GENERALE par L.Guibert | 1F |
| COMPUTING SYSTEM CONFIGURATION FOR FIGHTER AIRCRAFT FLIGHT CONTROL SYSTEMS by H.J.Kaul, H.Lüers and J.Rauch | 2 |
| A HIGHLY RELIABLE COMPUTER FOR AIRBORNE APPLICATIONS by M.Muenier | 3 |
| TESTING EMBEDDED SOFTWARE by M.Muenier | 4 |
| ADVANCED AVIONICS ARCHITECTURE — PAVE PILLAR by J.C.Ostgaard | 5 |
| DATA AND SIGNAL PROCESSING ARCHITECTURES FOR FUTURE AVIONICS by M.T.Michael | 6 |
| FIBER OPTIC BUSES AND NETWORKS FOR ADVANCED AVIONICS ARCHITECTURES by R.W.Uhlhorn | 7 |
| HIGH SPEED PARALLEL PROCESSING NETWORKS FOR ADVANCED ARCHITECTURES by D.R.Morgan | 8 |
| BIBLIOGRAPHY | B |

OVERVIEW AND INTRODUCTION
TO GCP LECTURE SERIES NO 158 ON
"COMPUTING SYSTEM CONFIGURATION FOR HIGHLY INTEGRATED
GUIDANCE AND CONTROL SYSTEMS"

by
IPA L. GUIBERT
Service technique des Télécommunications et des Equipements
aéronautiques
129, rue de la Convention
75731 PARIS CEDEX 15 - FRANCE

ABSTRACT

The avionics of modern aircraft are increasingly complex and perform an increasing amount of functions. As a result, configuration problems are assuming ever greater importance.

We should first define what we mean by system configuration. Once we start to do this, it becomes clear that because of the integration of functions it is impossible to isolate guidance and control systems if it is their architecture we are interested in.

Architecture concerns both equipment and systems or functions, which today include software, and this explains the variety of topics which will be dealt with in the papers to be given: design aids and methods, computers, data buses and software aids.

It is perhaps advisable to make a distinction between the functional architecture and the physical architecture of an aircraft. Given their importance and their effect on performance, reliability, survivability, maintainability and costs, and considering the complexity of the problems attaching to them, their design requires specific facilities.

The "Direction des Construction Aéronautiques" has joined forces with eight French companies to provide the aeronautics industry with an integrated workshop for the design of avionics systems. A brief description of the project is given.

INTRODUCTION

Fighter aircraft have to face an ever growing threat, both in terms of quantity and quality, on the ground as well as in the air, and probably shortly in space. In order to meet that threat, they are required to fulfil more and more functions under increasingly difficult conditions, while achieving ever higher performances.

As a result, aircraft acquisition and user cost continue to grow. Thus, in order to ensure the cost-effectiveness of new aircraft development, and because it has become difficult to financially support several programs, a trend towards the design multirole aircraft has recently increased.

The logical outcome is a further increase in the amount and complexity of the functions demanded.

All this is particularly true in avionics and weapon systems, which now handle the essential of the functions of an aircraft.

Thus, the acquisition cost of avionics (taken in its broadest sense to mean everything electronic in the plane) whose weight represents a fairly constant one tenth of the total empty weight of the aircraft, continues to increase in proportion to the overall cost, up to about 40% for next generation fighter aircraft, whereas its maintenance cost can form up to 80% of the overall one.

THE CONCEPT OF CONFIGURATION

That is the framework. In this context, what then is the part played by the computing system configuration of guidance and control systems?

First we need to define what that expression means.

Taking the above mentioned constraints and needs into account leads to an imbrication of the various functions performed by today's navigation and weapon systems.

The guidance of an aircraft, for instance, is involved in all stages of the mission, whatever it is, and must process data supplied by autonomous sensors (inertial units, terrain data memories, ...), radio-navigation units (GPS, ...), the radar (terrain profiles, location of targets, ...), the electronic counter-measures (for threat avoidance), or by the system itself (mission planning, tactical data, ...), etc.

This meshing of functions, which is necessary in order to keep avionics within reasonable volume and cost limits, and made possible by advances in microelectronics technology, leads to the withdrawal of the concept of sub-system.

Avionics can, in fact, be divided into sensors, core computing and terminals (mainly displays, controls or actuators). This does not necessarily mean that the system is centralized: core computing architecture may be spread, with certain processors remaining physically close to sensors or terminals. However, the whole core takes part in processing the data supplied by the sensors, and in generating those ones sent to the terminals.

The computing configuration of guidance and control systems cannot therefore easily be dissociated from the core avionics computing configuration, and this will be ever more true in the future, as we shall see from some of the lectures.

To simplify, computing systems are essentially composed with computers or processors, which are organized in such a way as to support functions, which are performed by software.

The configuration problems of such systems thus involve the design and definition of the computers, the overall organisation, or more precisely, the architecture, and the way in which the software is designed, using which tools and methodology.

OVERVIEW OF THE LECTURES

As you can see, the subject is vast. However, I thought that we should attempt to deal with each area, not, of course, in exhaustive manner, but by choosing certain important topics, so that the lectures will show the way things are evolving and identify the keys to the future, because it seems important to me to understand that all these aspects are linked, are parts of a whole.

Thus, the first lecture by Mr H. J. KAUL will deal with the question of flight control system design, and will describe tools in relation to a method.

We shall then go on to look at signal and data processors during the lectures by Mr M. T. MICHAEL and Mr M. MUENIER, in particular the Common Signal Processor, the 1750 A standard and a new high level processor, the CMF. Perhaps will they allow us to raise some traditional questions, such as should computers be standardized, will 32 bits supplant 16 bits, which instruction set (RISC or not RISC)?

The data buses, essential element of connection between the components of current systems, and from which considerable performance improvements are required for future systems, will be the subject of the lecture given by Mr R. UHLHORN, who will particularly address the subject of high speed optical buses.

We shall, of course, talk about software. It's a fundamental part of the system, and gives it life. Its volume is exponentially growing, as are the problems attaching to it. On existing aircraft, we can consider that the acquisition cost of the software for the central computer is half that of the computers themselves. But in operation, that ratio may reach 80% of the maintenance costs.

As future aircraft will include several million code lines, software quality and productivity are vital. Mr MUENIER will describe a tool for software testing, as part of a workshop about which I shall be giving you further informations later on.

We shall finally tackle the problems of avionics architecture with the lectures given by Mr J. C. OSTGAARD and Mr D. R. MORGAN on PAVE PILLAR. These statements will show how important this subject is, both in terms of its implications for avionics system design, and its effect on performance and costs.

THE ROLE OF ARCHITECTURE

The STANAG 3908 (Edition 1) defines architecture as: "in avionics, a representation of the hardware and software components of a system and their interrelationships, considered from the viewpoint of the whole system". As far as I am concerned, I would make a distinction as part of this definition, between the functional architecture and the physical one. To me, this would seem particularly important once we try, as the PAVE PILLAR concepts do, to standardize an advanced avionics architecture, that is, to make it common to several aircrafts, each with different missions.

Because of the diversity of missions, the system designers must be capable to implement the operational functions of an air vehicle (and only these ones, for cost reasons), in such a way as to best meet user requirements. From this functional point of view, every system should be specific, regardless of the level of standardization demanded for its components.

The functional architecture may be defined as the breakdown of the functions into functional modules, the organisation of these modules and of their interrelationships. The purpose is to get modules for parts of the software which can be realized by one single man and tested separately. To give you an idea, modules could contain up to 500 instructions maximum. The functional coherence is then ensured by means of a structured top-down breakdown approach, like the IDEF 0 method for example.

Physical architecture is related to the hardware, and may be defined as the organization and composition of the hardware and the interrelationships between its components required in order to support the functions.

Of course, in order for the hardware to be able to support all the functions, there must be compatibility between physical and functional architectures: they are therefore closely related, and should be designed together in order to produce a coherent system which meets requirements.

This is of great importance when we are attempting to standardize components and rules for the physical architecture. We shall see how the question has been resolved in the case of PAVE PILLAR.

I should like to add a rider on that one, if you will allow me. For a long time it has been assumed that technology could meet the increasing performance requirements made necessary by the developing threat. This has undoubtedly been the case, and I trust, still is. There is however a limit, which is imposed by cost. It appears today that even the most powerful nations cannot afford all the high performance systems which could give us a decisive advantage over the adversary.

As we shall see, optimum architectures result in considerable cost savings. This is the case for aircraft development and acquisition, where standardization approaches can prove highly beneficial. It also applies to life cycle cost, which is an attractive prospect, as rising maintenance costs eat into the funding available for new developments. What is more, technology and performance being equal, architecture optimization produces substantial gains in terms of availability, survivability, maintainability, whose effects come as an added bonus to the reductions in maintenance cost achieved by the technology (higher MTBF) and the architecture studies (reduction in the amount of maintenance levels and quantity of spare parts in stock, for instance).

As you can see, these questions are fundamental to our ability to produce efficient systems.

A SYSTEM APPROACH

We have just noticed that system design and architecture problems are vital. They are also difficult to master. Just think of the number of 500 lines modules that may be contained in a software program with several million instructions! The work involved in designing modern fighter avionics far exceeds human capabilities. We must therefore use stringent methods, backed up by efficient computer techniques.

The French aeronautics industry has undertaken in the last years a considerable effort, with support by the Ministry of Defence, in order to introduce such facilities for the whole system design cycle.

This step appears to me as exemplary, and I should like to describe its main principles. It is called the ITI (*Intégration du Traitement de l'Information*) program (Data processing integration) which is concerned. It is headed by the Direction des Constructions Aéronautiques (DCAé), with the assistance of eight companies: two aircraft manufacturers, AEROSPATIALE and AMD-BA, and six equipment manufacturers, CROUZET, SAGEM, SFENA, SFIM and THOMSON-CSF.

ITI offers a solution which meets a number of needs for the development of future avionics systems.

The problems are the following.

- The growing importance of software, which is also becoming increasingly complex, as a result of systems integration, and because of the fact that it has to handle most of the modifications made during development. This product must be of maximum quality, which means channeling the creativity of those responsible for its design, by means of rigorous methods. The apparent ease with which modifications can be made is, in fact, a major risk for quality. On the other hand, and everyone finds this worrying, software is not experiencing the same productivity growth as the other activities, at the very time when its volume is increasing exponentially.

- The coexistence of various versions of the same system, whether for successive versions of the same aircraft incorporating new capabilities, or those intended for different clients.

- The growing proportion of on board equipment (for instance, 60 equipment items of 35 different types, and 14 types of option for the AIRBUS A 320).

- The systems complexity, due to the performances specifications, requiring leading edge technology solutions, and the high number of modifications, lead to higher cost and longer delivery times.

Three major needs emerge from these considerations.

- We must have complete technical mastery of development. The process must therefore be automated.

- We must control costs and implementation times, which means improving our forecasts, providing efficient project management, and reducing or better eliminating risks.

- We must improve software quality and productivity. This means enforcing strict design rules. This requires too powerful software specification, production and debugging aids.

Moreover, these four points lead naturally to closer cooperation between the various companies working on a given project: the amounts of data exchanged are enormous. For example, for one version of the MIRAGE 2000, the exchanged documents (apart from user's documentation as such) represent a stack of paper more than eight meters high. Now, if the tools used by the different parties involved are not harmonized, this information has to be sorted more or less manually at each stage before being processed, which is a considerable waste of time.

In order to overcome these problems, ITI has laid down the following aims:

- to facilitate communications between the companies working on the same project. This is made possible *inter alia* by the adoption of a common work methodology, by automatic document processing and by electronic mailing for transfer of documents in a form directly usable by the addressee;

- to provide a system design aid, using computers for design, specification, definition;

- to provide a software development aid, using computers for design, definition, encoding and testing. One of the testing aids used, IDAS, will be discussed by Mr MUENIER;

-to assist with integration and validation of software and systems.

Interoperability problems also need to be taken into account. Since the design aids are to be used by various partners, they must be compatible with any computer configuration, provided it meets certain standards.

Finally, security objectives had to be defined: protection of software against the supply of erroneous data and incorrect handling, automatic error detection with tell-back, etc.

To attain these various ends, an integrated workshop known as SDA (Avionics Development System) was set up.

It is an integrated workshop in the sense that the aids that it comprise can communicate with each other, and exchange the results of the tasks in which they are involved. The structure of the SDA is based on the industrial organisation of a project, and is on three levels. The first is that of communications between the different companies participating to a program. The second is that of the manufacturer, who has his own computing facilities: it is the Manufacturer Development System (SDI-Système de Développement Industriel). The third is that of the individual user in his specific work situation inside his company: it is the Personal Development System (SDP-Système de Développement Personnel).

Communications between the SDP's belonging to an SDI on the one hand, and between the SDI's within an SDA in the other, are provided by an access structure.

This access structure combines the mechanisms and aids which are common to all SDI's.

Examples of common aids are: the operating system, the composer, the administrator, the configuration management, the object management system, etc.

On this basis, each SDP, depending on the need, integrates general tools (documentation, project management, quality, etc...) and specific tools, for the design and elaboration of the functional architecture (identification of functions and interfaces), definition of the avionics system, specification of its components, software definition and design, software production activities (encoding, production of executables, individual testing, using high order languages such as ADA and LTR 3), and integration and validation, using a dynamic test aid.

CONCLUSION

Clearly, the subject matter of this Lecture Series is vast. The concept of system configuration covers a whole series of activities, all of which have their part to play in the production of high performance aircrafts. One cannot hope to treat such a subject exhaustively. I merely hope that by the end of the final lecture, you will be aware of the importance of these problems though your attendance is already proof of that, and that you will have a little clearer perception of the challenges of the near future.

BIBLIOGRAPHY

ITI Summary Report No ITI-102/86

**INTRODUCTION ET PRESENTATION GENERALE
DE LA LECTURE SERIES N° 158 SUR
"CONFIGURATION INFORMATIQUE POUR SYSTEMES DE GUIDAGE ET PILOTAGE
HAUTEMENT INTEGRES"**

par

IPA L. GUIBERT

Service Technique des Télécommunications et des Equipements aéronautiques
129, rue de la Convention
75731 PARIS CEDEX 15. FRANCE

RESUME

L'avionique des avions de combat modernes est de plus en plus complexe et remplit des fonctions toujours plus nombreuses. Pour cette raison, les problèmes liés aux configurations sont de plus en plus importants.

Il faut dans un premier temps définir ce que l'on entend par configuration des systèmes.

Ce faisant, il apparaît clairement que du fait de l'intégration des fonctions, il est impossible d'isoler les systèmes de guidage et de pilotage dès lors qu'on s'intéresse à leur architecture.

L'architecture concerne aussi bien les équipements que les systèmes ou les fonctions (donc, aujourd'hui, les logiciels). C'est pourquoi des thèmes divers seront abordés au cours des différents exposés : méthodes et outils de conception, calculateurs, bus de données, outils logiciel.

Il est peut-être bon de différencier sur un avion l'architecture fonctionnelle et l'architecture physique. Compte tenu de leur importance et de leur incidence sur les performances, la fiabilité, la survivabilité, la maintenabilité, les coûts, et du fait de la complexité des problèmes qui leur sont liés, leur élaboration nécessite des outils adaptés.

La Direction des Constructions Aéronautiques et huit sociétés françaises ont fait un effort particulier pour rendre disponible pour l'industrie aéronautique un atelier intégré pour la conception des systèmes avioniques. Il est brièvement décrit.

INTRODUCTION

Les avions de combat ont à faire face à une menace qui ne cesse de croître, en quantité et en qualité, que ce soit au sol ou dans les airs, bientôt même probablement dans l'espace. Pour s'adapter à cette menace, on leur demande de remplir des fonctions de plus en plus nombreuses, dans des conditions plus difficiles et avec des performances toujours améliorées.

Cela étant, le coût des avions ne cesse d'augmenter, aussi bien à l'acquisition qu'en utilisation opérationnelle. Si bien que pour rentabiliser le coût de développement des avions nouveaux, et parce qu'il est devenu difficile de supporter la charge financière de plusieurs programmes parallèles, une tendance à concevoir des avions multi-rôles s'est affirmée ces derniers temps.

La conséquence, c'est encore l'augmentation du nombre et de la complexité des fonctions.

Tout cela est particulièrement vrai pour l'avionique et les systèmes d'armes, qui remplissent désormais l'essentiel des fonctions dévolues à l'avion.

C'est ainsi que pour une masse qui représente de façon constante environ le dixième de la masse à vide de l'avion, le coût d'acquisition de l'avionique (au sens large: tout ce qui est électronique dans l'avion) ne cesse d'augmenter en proportion du coût global, jusqu'à environ 40% pour les avions de combat de la prochaine génération, tandis que son coût de maintenance peut représenter jusqu'à 80% du coût de maintenance global.

LA NOTION DE CONFIGURATION

Voilà le cadre. Dans ce contexte, quel est le rôle de la configuration informatique des systèmes de pilotage et de guidage?

Avant tout, il est nécessaire de s'entendre sur ce que recouvre cette expression.

La prise en compte des contraintes et des besoins rappelés ci-dessus conduisent à une imbrication des différentes fonctions réalisées par les systèmes de navigation et d'armement modernes.

Le guidage d'un aéronef, par exemple, intervient dans toutes les phases d'une mission, quelle qu'elle soit, et doit traiter des informations qui proviennent aussi bien de capteurs autonomes (centrales de navigation, fichiers de terrain,...) que de capteurs de radio-navigation (GPS,...), du radar (profils de terrain, position des cibles,...), des contre-mesures (pour l'évitement des menaces), du système (plan de mission, informations tactiques,...), etc.

Cette imbrication des fonctions, nécessaire pour maintenir l'avionique à un volume et un coût raisonnables, et permise par les technologies de la micro-électronique, conduit à la disparition de la notion de sous-système.

L'avionique se répartit en fait entre des capteurs, un cœur informatique, et des terminaux (visualisations, commandes, ou actionneurs, essentiellement). Cela ne signifie pas qu'il y ait nécessairement centralisation: le cœur informatique peut être conçu selon une architecture répartie, certains processeurs pouvant rester physiquement "proches" des capteurs ou des terminaux. Mais l'ensemble du cœur participe souvent à l'exploitation des informations provenant des capteurs et à l'élaboration de celles envoyées aux terminaux.

La configuration informatique des systèmes de pilotage et de guidage est donc difficilement dissociable de celle du cœur de l'avionique, et ce sera de plus en plus le cas dans le futur, comme nous le verrons au cours de quelques exposés.

Un système informatique, c'est essentiellement, en schématisant, des calculateurs ou des processeurs organisés de telle sorte qu'ils puissent supporter des fonctions qui sont réalisées par du logiciel.

La configuration d'un tel ensemble concerne donc la conception: et la définition des calculateurs, l'organisation de l'ensemble ou plus précisément l'architecture du système, et la façon dont est conçu le logiciel, selon quelle méthodologie et avec quels outils.

GENERALITES SUR LES CONFERENCES

On le voit, le sujet est vaste. Cependant, j'ai pensé qu'il fallait essayer d'aborder chaque thème, non pas bien sûr pour les présenter de façon exhaustive, mais en choisissant quelques sujets importants où les exposés montreront dans quelles directions évoluent les choses et quelles sont les clés du futur, parce qu'il me paraît important de prendre conscience que toutes ces choses sont liées, sont les parties indissociables d'un tout.

Ainsi le premier exposé, de M. H. J. KAUL, abordera les problèmes de conception des systèmes de commande de vol et décrira des outils liés à une méthode.

Nous nous intéresserons au cours des exposés de MM. M. T. MICHAEL et M. MUENIER aux processeurs de traitement de données et de signal, en particulier au CSP, au standard 1750 A et à un nouveau processeur de haut niveau, le CMF. Peut-être nous permettront-ils d'aborder des questions désormais traditionnelles: faut-il standardiser des calculateurs, le 32 bits va-t-il supplanter le 16 bits, quel jeu d'instructions: RISC ou non?

Les bus de données, élément essentiel de liaison entre les composantes des systèmes actuels et auxquels des performances en augmentation notable sont demandées pour les systèmes futurs, feront l'objet de l'exposé de M. R. UHLHORN, qui abordera en particulier les bus optiques à grand débit.

Le logiciel bien sûr ne peut être absent ici. Constituant essentiel du système, auquel il donne vie, son volume croît de façon exponentielle, et dans le même temps les problèmes qui lui sont attachés. Sur des aéronefs existants, on peut considérer que le logiciel du calculateur central coûte à l'acquisition la moitié des calculateurs eux-mêmes. Mais en utilisation, la proportion peut aller jusqu'à 80% des coûts de maintenance.

Pour les aéronefs futurs, qui intégreront plusieurs millions de lignes de code, la qualité et la productivité en matière de logiciel sont des points de passage obligés. M. MUENIER nous parlera d'un outil de test de logiciel, qui sera intégré dans un atelier dont j'aurai l'occasion de vous dire quelques mots par la suite.

Nous nous intéresserons enfin aux problèmes d'architecture avionique lors des exposés de MM. J. C. OSTGAARD et D. R. MORGAN sur PAVE-PILLAR. Ces présentations montreront l'importance du sujet, tant par ses implications sur la conception des systèmes avioniques que par ses conséquences sur les performances et les coûts.

LE ROLE DE L'ARCHITECTURE

Le STANAG 3908 (Edition 1) définit l'architecture comme étant "en avionique, une représentation des composants matériels et logiciels d'un système et de leur relations, considérés du point de vue du système global". Pour ma part, je proposerais de distinguer dans le cadre de cette définition entre l'architecture fonctionnelle et l'architecture matérielle. Cela me paraît particulièrement important dès lors que l'on cherche, comme le proposent les concepts de PAVE-PILLAR, de standardiser une architecture avionique avancée, c'est-à-dire de la rendre commune à plusieurs aéronefs dont les missions sont différentes.

La diversité des missions impose en effet que le concepteur du système puisse implanter les fonctions de l'avion (et seulement celles-là, pour des raisons de coût) pour répondre au mieux aux besoins exprimés par l'utilisateur. De ce point de vue fonctionnel, chaque système doit pouvoir être spécifique, quelque soit le niveau de standardisation désiré pour ses composants.

L'architecture fonctionnelle pourrait être définie comme étant le découpage des fonctions en modules fonctionnels, l'organisation des modules fonctionnels et de leurs relations. Aujourd'hui, cela concerne essentiellement le logiciel d'application. Le but est d'obtenir des modules correspondant à des parties de logiciel réalisables par un seul individu, et testables isolément, pour fixer les idées, comprenant un maximum de 500 lignes de code. La cohérence fonctionnelle est alors assurée par une décomposition selon des approches top-down structurées, par exemple selon des méthodes du type IDEF-0.

L'architecture matérielle concerne le hard, et peut être définie comme l'organisation et la composition du matériel et des relations entre ses éléments en vue de pouvoir supporter les fonctions.

Bien entendu, pour que les matériels soient capables de supporter toutes les fonctions, il faut assurer la compatibilité des architectures fonctionnelles et matérielles: elles sont donc très liées l'une à l'autre et doivent être définies ensemble dans le but de réaliser un système cohérent répondant aux besoins.

C'est particulièrement important quand on cherche à standardiser des éléments et des règles pour l'architecture matérielle, et nous verrons comment cette question a été résolue dans le cadre de PAVE-PILLAR.

Encore une réflexion, si vous le permettez, sur ce point. On a longtemps considéré que la technologie permettait de répondre aux besoins croissants en performance dus à l'évolution de la menace. Cela est sûrement vrai et, je l'espère, se trouve vérifié jusqu'à présent. Mais il y a une limite, qui est constituée par les coûts. Il apparaît aujourd'hui que les plus grandes nations ne peuvent payer tous les systèmes très performants, qui pourraient à coup sûr nous donner un avantage certain sur l'adversaire.

La définition d'architectures optimales permet, comme nous le verrons, d'abaisser notablement le coût des systèmes. C'est le cas au niveau du développement et de l'acquisition des aéronefs, et là, les approches de standardisation peuvent être très bénéfiques. C'est aussi le cas pour le coût de cycle de vie, ce qui est particulièrement intéressant, puisque l'augmentation des coûts de maintenance grève sensiblement les capacités de financement des développements nouveaux. Or l'optimisation des architectures apporte, à technologie et performances égales, des gains substantiels en matière de disponibilité, survivabilité, maintenabilité, dont les effets s'ajoutent aux diminutions des coûts de maintenance obtenues par la technologie (augmentation du MTBF,...) et les études d'architecture (diminution du nombre des niveaux de maintenance, du volume des stocks de rechanges, par exemple).

On le voit, ces problèmes constituent une clé fondamentale pour notre capacité à produire des systèmes performants.

UNE APPROCHE SYSTEMIQUE

Nous venons de constater que les problèmes de conception des systèmes, d'architecture, sont primordiaux. Ils sont aussi difficiles à maîtriser. Songez au nombre de modules de 500 lignes que peut compter un logiciel de plusieurs millions d'instructions. Le travail à accomplir pour concevoir l'avionique des aéronefs modernes dépasse les capacités humaines. Il est donc nécessaire d'utiliser des méthodes rigoureuses, sous-tendues par des outils informatiques performants.

L'industrie aéronautique française a engagé ces dernières années un effort important pour se doter d'outils couvrant tout le cycle de conception d'un système, avec le soutien du Ministère de la Défense.

Cette démarche me paraît exemplaire et, si vous le permettez, j'aimerais vous en exposer les principes. Il s'agit du programme ITI (Intégration du Traitement de l'Information), mené par la Direction des Constructions Aéronautiques (DCAé) auprès de huit industriels: deux avionneurs AEROSPATIALE et AMD-BA, et six équipementiers, CROUZET, SAGEM, SFENA, SFIM et THOMSON-CSF.

ITI propose une solution pour satisfaire un certain nombre de besoins pour le développement des systèmes avioniques futurs.

Les problèmes sont les suivants.

-L'importance croissante des logiciels, par ailleurs de plus en plus complexes de par l'intégration des fonctions et parce que l'on reporte sur lui les modifications nécessaires au cours du développement. Il faut assurer à ce produit une qualité maximum, et donc canaliser par des méthodes rigoureuses la créativité des personnes responsables de sa réalisation. La facilité apparente de faire des modifications est en effet un risque majeur de non qualité. D'autre part, et tout le monde s'en inquiète, on n'observe pas la même croissance de productivité en matière de logiciel que pour les autres activités, alors que les volumes augmentent de façon exponentielle.

-La coexistence de nombreuses versions de systèmes, que ce soit les versions successives, incorporant des nouvelles capacités, d'un même aéronef ou celles destinées à des clients différents.

-La part grandissante des équipements embarqués (par exemple pour l'A 320, 60 équipements de 35 types différents, et 14 types d'optionnels).

-La complexité des systèmes, due aux performances demandées, qui conduisent à mettre en oeuvre des solutions à la pointe de la technologie, et le nombre des modifications à prendre en compte, font augmenter les coûts et les délais.

Il en découle trois grands besoins.

-Il faut garantir la maîtrise technique du développement. Il faut donc automatiser le processus.

-Il faut maîtriser les coûts et les délais, et pour cela, améliorer les prévisions, assurer un suivi de projet efficace, et réduire, et essayer d'éliminer, les aléas.

-Il faut améliorer la qualité et la productivité en matière de logiciel. Cela implique d'imposer des règles strictes de conception. Cela demande aussi des outils puissants d'aide à la spécification, à la réalisation et à la validation des logiciels.

De plus, ces quatre points induisent naturellement un accroissement des relations entre les divers coopérants d'un programme: les volumes de données échangées deviennent énormes. Par exemple, pour une version du MIRAGE 2000, l'ensemble des documents échangés (hors documentation) représente une pile de papier de plus de 8 mètres de hauteur. Or, s'il n'y a pas harmonisation des outils des intervenants, ces informations doivent être ressaisies plus ou moins manuellement à chaque étape du processus, avant leur exploitation, ce qui représente une perte de temps inutile.

Pour parvenir à remédier aux problèmes, ITI s'est fixé les objectifs suivants:

-faciliter la communication entre les industriels intervenant dans un projet. Cela est rendu possible entre autres par l'adoption d'une méthodologie de travail commune, par la réalisation de façon automatique de documents informatisés, et par une messagerie informatique permettant l'échange de ces documents sous une forme directement exploitable par le destinataire;

-apporter une aide à la conception des systèmes, grâce à des outils informatiques pour la conception, la spécification et la définition;

-apporter une aide au développement des logiciels grâce à des outils pour la définition, la conception, le codage et le test. Un outil de test retenu est IDAS, qui fera l'objet du deuxième exposé de M. MUENIER;

-aider à l'intégration et à la validation des logiciels et des systèmes.

De plus sont pris en compte des objectifs de portabilité. Les outils, devant être utilisés par plusieurs intervenants, doivent pouvoir être implantés sur chaque configuration informatique, à condition qu'elle satisfasse certains standards.

Enfin ont été définis des objectifs de sécurité: protection des logiciels contre la fourniture d'informations erronées et les mauvaises manipulations, détection automatique d'erreurs avec compte rendu, etc.

Pour atteindre ces divers objectifs, un atelier intégré, appelé SDA (Système de Développement d'Avionique), a été défini.

C'est un atelier intégré en ce sens que tous les outils qu'il comporte peuvent communiquer entre eux, et s'échanger les résultats des tâches auxquels ils concourent.

L'organisation du SDA est calquée sur l'organisation industrielle d'un projet, et comprend trois niveaux. Le premier est celui des communications entre les divers industriels participant au programme. Le deuxième niveau est celui de l'industriel qui dispose de moyens informatiques propres: c'est le Système de Développement Industriel, SDI. Le troisième est celui de l'utilisateur individuel dans son contexte de travail à l'intérieur de son entreprise: c'est le Système de Développement Personnel, SDP.

Les communications entre les SDP appartenant à un SDI d'une part, et entre les SDI reliés au sein d'un SDA d'autre part, sont assurées par une structure d'accueil.

Cette structure d'accueil rassemble les mécanismes et outils communs à tous les SDI.

Les outils communs sont par exemple: le système d'exploitation, le compositeur, l'administrateur, la gestion de la configuration, le système de gestion d'objet, etc.

Sur cette base, chaque SDP intègre en fonction de ses besoins des outils généraux (documentation, gestion de projet, qualité, etc) et des outils spécifiques, pour la conception et l'élaboration de l'architecture fonctionnelle (identification des fonctions et des interfaces), la définition des systèmes, la spécification de leurs composants, la définition et la conception du logiciel, les activités de sa réalisation (codage, production d'exécutable, tests unitaires, avec des langages de haut niveau, comme ADA et LTR 3), l'intégration, et la validation à l'aide d'un outil de test dynamique.

CONCLUSION

Il est clair que le sujet de ce cycle de conférences est vaste. La notion de configuration des systèmes recouvre toute une série d'activités, qui concourent chacune d'elle à la réalisation d'aéronefs performants. On ne peut espérer aborder chaque sujet de manière complète. J'espère seulement que demain, après le dernier exposé, vous aurez conscience de l'importance de ces problèmes (mais votre présence ici prouve que c'est déjà le cas), et aurez une vision un peu plus nette des challenges du proche avenir.

REFERENCES

Rapport de synthèse des travaux ITI. N° ITI-102/86

COMPUTING SYSTEM CONFIGURATION FOR FIGHTER AIRCRAFT FLIGHT CONTROL SYSTEMS

by

H.J. Kaul, H. Lüers, J. Rauch
Messerschmitt-Bölkow-Blohm GmbH
Aircraft Division
8000 München 80, Postfach 801160

ABSTRACT

Flight Control Systems (FCS) for present and future Fighter Aircraft developments are based on three basic technologies, CCV-ACT-digital signal processing. These technologies have opened a new degree of freedom for optimizing overall weapon system performance by extending the requirements to be implemented by the FCS. The computing subsystem of the FCS is the key element of the FCS by which this performance optimization can be achieved. Digital FCS for fighter aircraft in service or under development exhibit a basically common architecture (static parallel redundancy, centralized heavily burdened computing subsystem) which has continuously evolved since the early days of CCV and ACT activities. The objectives of the paper are to relate this classical architecture to the advancing requirements and the new emerging technologies and to analyse its potentials for future developments. From an airframe manufacturers point of view the paper will

- o attempt a critical review of the interrelationship between weapon system performance and design considerations involved in the development of the computing subsystem of the FCS (present and future).
- o address the problem of restrictions imposed upon the application of advanced computer hardware/software and micro-electronic technologies by FCS specific design considerations (present).
- o identify the need to overcome state of the art FCS architectures in order to fully utilize the potential of emerging technologies for the benefit of weapon system performance through an advanced design of the computing sub-system of the FCS (future).

LIST OF ABBREVIATIONS

| | |
|-------|---|
| ACT | Active Control Technology |
| A/C | Aircraft |
| ADS | Air Data System |
| ADT | Air Data Transducer Unit |
| ASIC | Application Specific Integrated Circuit |
| AVS | Avionic System |
| BC | Bus Controller |
| BIT | Built-In-Test |
| CCV | Control Configured Vehicle |
| CBIT | Continuous Built-In-Test |
| C.G. | Centre of Gravity |
| CIFU | Cockpit Interface Unit |
| CBIT | Continuous Built-In-Test |
| CISC | Complex Instruction Set Computer |
| CMAM | Custom Monolithic Analogue Microcircuit |
| CPU | Central Processing Unit |
| DARPA | Defense Advanced Research Projects Agency |
| DDV | Direct Drive Valve |
| DSP | Digital Signal Processing |

| | |
|------|--|
| EAP | Experimental Aircraft Programme |
| EFA | European Fighter Aircraft |
| EHSV | Electro Hydraulic Servo Valve |
| EMP | Electromagnetic Pulse |
| EMI | Electromagnetic Interference |
| FCS | Flight Control System |
| FCC | Flight Control Computer |
| FH | Flying Hours |
| HQ | Handling Quality |
| IBIT | Initiated Built-In Test |
| ICD | Interface Control Document |
| IMS | Inertial Measurement System |
| IMU | Inertial Measurement Unit |
| INR | Initial Nuclear Radiation |
| I/O | Input/Output |
| IPSE | Integrated Project Support Environment |
| LRI | Line Replaceable Item |
| LRM | Line Replaceable Module |
| LSI | Large Scale Integration |
| LVDT | Linear Variable Differential Transformer |
| MBB | Messerschmitt-Bölkow-Blohm GmbH |
| NEMP | Nuclear and Electromagnetic Pulse |
| PFC | Pre-Flight Check |
| RFI | Radio Frequency Interference |
| RISC | Reduced Instruction Set Computer |
| RT | Remote Terminal |
| RVDT | Rotary Variable Differential Transformer |
| SDI | Strategic Defense Initiative |
| VLSI | Very Large Scale Integration |

1. INTRODUCTION

Present and future developments in the field of combat aircraft are based on the principles of CCV technology. The design process of the airframe can be carried out without a too restrictive need for compromise with stability and control requirements. The advantage for a fighter to be gained from this approach are focussing on enhanced manoeuvrability, high performance and excellent and full carefree handling and control of the aircraft. This leads to configurations with a high level of longitudinal instability together with a partial instability in the lateral axis which in consequence dictates a need for a full-time authority fly-by-wire system.

Major developments like CCV-F104, Jaguar FBW, F16, F18 and EAP [1,2] have brought the technology for such FCS's to a highly mature state. Advances in Active Control Technology (ACT) and the increasing performance of digital signal processing have opened a new degree of freedom for optimizing overall weapon system performance through the design of the FCS. However, one has to recognize that the systems in service so far have utilized only a small segment of this potential. Recent Experimental programs like X-31 and EAP are gradually extending into this potential. The maturity mentioned above means that we have learned to cope with associated safety issues within the constraints of available implementation technologies. Keeping this experience as an essential basis, new developments can extend their functional performance [3].

In the area of CCV and the associated FCS technologies, MBB has an accumulated experience over a wide range of activities (see Fig. 1) starting from the CCV-F104 program through to the beginning of the development phase for the European Fighter Aircraft (EFA) program and the X-31 program.

For a fighter aircraft (A/C) developed along the line of a CCV approach, the FCS performs a twofold function.

- o It forms an integral unit together with the airframe - the weapon system carrier - optimized with respect to manoeuvrability and performance
- o It is an integral part of the avionic suite loaded into the carrier with the aim to
 - assist the pilot in performing his tactical task and
 - provide semi- and fully automatic modes to enhance weapon system effectiveness

The functions to be performed by the flight controller are summarized in Fig. 2.

The specification of a full authority, full time fly-by-wire control system architecture i.e. with no facility for reversion to mechanical controls is a technology integration task. State-of-the-art assessments and trends in the underlying computing, sensing and actuation areas have to be performed to select from a number of design alternatives. Since this paper concentrates on FCS computing system configuration aspects only, all sensing and actuation design considerations are addressed from the computing point of view. The computing system (see Fig. 3) comprises the following functional elements:

- o Processing of sensor raw data
- o Processing of signals from/to related cockpit elements
- o Flight controller algorithms
- o Actuator loop closures
- o Signal transmission within the computational system and between the computational system and the sensors, FCS related cockpit elements and actuators
- o Communication with general systems and the avionic system
- o FCS failure tolerance and reconfiguration capability
- o Control and management of all tasks within the computational system

All functional elements outside this definition will only be addressed with respect to their impact on the functional elements listed above. This rather broad definition supports a line of discussion starting from functional aspects reflecting the requirements down to implementation issues. The following sections will outline from an airframe manufacturers point of view the FCS design objectives and constraints in the light of (present and future) computing system design considerations, present restrictions imposed upon the application of advanced computer hardware/software and the potential of emerging technologies for the benefit of weapon system performance.

2. ANALYSIS OF REQUIREMENTS

2.1 Airframe Related Requirements

The first task a FCS has to perform in a CCV-A/C is to ensure stability and thus providing a basis for acceptable control characteristics. CCV technology does not mean that the aerodynamicist has unrestrained freedom in defining the aircraft configuration. The control system imposes restrictions on the allowable instability levels.

In the early stage of the definition of the airframe, a simple model of the envisaged control system must be defined on the basis of past experience data (see Fig. 4). Closed loop simulation work with preliminary aerodynamic data representing the airframe and the controller model are used to define acceptable

instability levels. These simulations shall ensure that the stability requirements defined in MIL-F-9490D will be met by the augmented airframe in the following areas.

- o in the whole envisaged flight envelope
- o for all planned configurations (stores)
- o for all c.g. locations

Once the process of defining the aerodynamic configuration has converged to a configuration freeze, the control system model and its parameters become the dominant requirements for the FCS design. During the subsequent development process, the compliance of the implementation elements with this model has to be monitored very closely.

Since the control system model leading to this prime set of requirements has to be defined in an early stage of a project, the data used must have a level of confidence which is dependent on development risk accepted for the program. Taking a medium level of development risk, this means that the parameters representing system components (sensors, computers, actuators, etc.) reflect technologies which have been proven in at least a laboratory environment approximately 5 years prior to first flight.

An important step is to define a subset of the above control system model which represents the minimal configuration of the control system required to provide stability levels within restricted flight and manoeuvre envelopes sufficient to recover the aircraft in case of system failures, i.e. Operational State III as defined in MIL-F-9490D. This minimal configuration defines the safety critical functional path of the system.

Another important function of the control system related to the airframe is the automatic control of the aerodynamic configuration. A typical example is given in Fig. 5 to demonstrate the multiple use of the aerodynamic control surfaces. Besides stabilization and control they are used for trim and performance optimization and control of aerodynamic configuration to improve on lateral directional stability at high incidence providing satisfactory levels of spin departure resistance [2].

2.2. Handling Performance and Automatic Modes

The outcome of the airframe related requirements for the control system forms the basis for the subsequent work. The major impact of this work on the controller requirements and consequently on the computing system is through

- o an extension of the required measurement values for pitch angle (θ) and bank angle (ϕ) both needed for g-compensation
- o the definition of the scheduling parameters required by the control laws.

A schematic block diagram of the resulting control law structure is given in Fig. 6.

Carefree manoeuvring of the aircraft is one of major objectives. This rather vague expression summarizes the functions which have to be incorporated to ensure an automatic way that the safe flight envelope of the aircraft is respected wherever possible and without compromising manoeuvrability. Though raising complicated issues when designing the associated control law algorithms, its impact through the functional requirements on the computing subsystem is relatively simple. All measured parameters being subject to an automatic control are already contained in the aforementioned requirements such as incidence and sideslip, vertical acceleration n_z and airspeed. The additional requirement is for

parameters needed to define the actual limitations such as aircraft mass, c. g. locations and external stores configuration. These parameters have to be made available to the flight control system through communication with other aircraft systems.

The problems associated with the topic of automatic modes are beyond those of conventional basic Autopilot Modes. They reflect the important step from pilot interpreted integration to the next higher level of automated integration, (see Fig. 7). An example is the integration of FCS and engine control.

The complete set of functional elements constituting such automatic modes are not attributable to the flight controller alone. It is a task for overall avionic system design to resolve the total function to subsystem level whereby the flight control system is regarded as a subsystem. The impact on the computing section of today's FCS is through software requirements and requirements for the communication system to avionic subsystems.

2.3 Fault Tolerance Requirements

The functions of the flight controller and their performance level are dependent on the system failure state (Table 1). The definition of operational states IV and V was chosen to cover the peculiarities of flight control systems for highly unstable airframes. The detailed reasoning behind this is beyond the scope of this paper but it simply says that you can either fly the aircraft within the performance of state III or you lose it. The relation between the operational states and the safety and mission requirements is shown in Fig. 8.

The current design objective is that aircraft losses due to hazardous technical failures in the FCS alone shall not exceed $1 \times 10^{-6}/FH$. The total aircraft loss rate attributable to both FCS failures alone and FCS failures in combination with failures in other aircraft subsystems (including total loss of electrical and hydraulic supplies affecting the FCS) shall not exceed $2 \times 10^{-6}/FH$. This requirement is harmonized with the overall aircraft loss rate and connected to the contribution of other vital A/C systems such as engines, hydraulic supplies, etc. It seems to be a valid assumption that this requirement will not go up for future fighter A/C developments.

The electrical part of the FCS which uses primary inputs (i.e. pilot commands and aircraft rate and acceleration sensors) and drives the first stages of the primary control actuators shall be capable of continuing to operate after any single failure without significant loss of performance, furthermore sequential double failures shall be survived, albeit with some loss of performance. Secondary functions which optimize the performance of the aircraft but may fail passively can be of a lower redundancy level provided the overall integrity requirement for the system is met.

These requirements can usually be satisfied by conventional triplex or quadruplex systems whereby the lanes are determined by the redundant computing elements of the safety critical functional path, the choice depending on whether a 100 % "two fail-operate" requirement is imposed.

The contribution of the FCS to aircraft mission failure rate (i.e. to operate the A/C as a weapon system) shall not exceed 2 per 1000 missions excluding hydraulic and electrical supplies. One or a combination of the following conditions are identified to cause an aborted mission:

- o the degradation of the redundancy level of the FCS is such that a next failure could result in the loss of the A/C;
- o the HQ (MIL-F-8785) are degraded down to level 3;

- o the degraded performance is not adequate for any remaining part of the mission;
- o the Pre-Flight Check (PFC) results in a NO-GO condition

A mission may be reduced if the failure situation of the FCS results in one or a combination of the following conditions:

- o The HQ are degraded down to Level 2;
- o The degraded performance is not adequate for all parts of the mission.

For the assessment of mission reliability an approximative method is applied, which combines mission rates and times of a mission scenario in order to obtain a mission reliability figure.

The defect rate of the FCS shall not exceed 20 per 1000 FH and does not include hydraulic and electric supplies.

The apportionment of reliability figures to electronic equipments is based on the following considerations:

- o the requirements to be fulfilled for the total system are the starting point
- o the relative complexity of the LRIs defines apportionment of the total figures
- o the figures derived in this manner are to be compared with available information about failure/defect rates of comparable equipments used in other projects
- o relationship of defect rate to failure rate
- o relation between flying hours and operating hours
- o a "reserve factor" is to be applied in order to counter for possible non-fulfillment of specified figures, i.e. the specified figures are made more stringent than the apportioned figures.

2.3.1 Redundancy Management Requirements

The FCS has to possess redundancy management techniques capable of providing optimum failure survivability via detection and isolation of failed components and reconfiguring the remaining healthy components to provide the maximum level of aircraft safety and the highest probability of mission completion. A safe, dependable failure detection and identification scheme is necessary for detecting and isolating failures in FCCs, sensors, data links, actuation systems, electrical power supplies, hydraulic power supplies in such a manner that it does not degrade the capability to detect and isolate subsequent failures. If a failure has been detected, reconfiguration to the next lower redundancy level of the affected function shall be performed and no unacceptable transient be caused due to reconfiguration. Once a degradation of redundancy level has taken place, the system shall not automatically upgrade to the next higher level if the failure disappears. System degradation by nuisance reconfiguration due to false alarms must be considered when system performance is being evaluated.

The FCS shall be safely upgraded by issuing a pilot initiated reset which shall lead to a reconfiguration state in which attempts shall be made to reset all

faulty elements after appropriate tests. Only a single failure in a sensor set can be cleared if the sensor successfully passed the reset test and no dormant failure or singularity shall arise as a result of this action. Only one off-line lane can be upgraded at a time in a similar procedure. This capability will improve the mission completion rate.

Voter- and failure detection and diagnostic algorithms have to be a function of the current sensor set configuration as defined by a consistent view of all non-faulty channels and must be capable of meeting the failure transient requirements. Means have to be provided to cope with malicious faults in the data replication step of time-varying data (input sensors) in such a manner, that the likelihood of not reaching exact mutual agreement in all non-faulty lanes at the voter output is compatible with the overall integrity objective of the FCS. Results of the input voting plane which consists of analogue, discrete and digital input signal management form the values which are supplied to other tasks (e.g. control laws etc.). The signal management of control law demands shall produce output demands that do not drift apart greater than a constant tolerance with respect to one another in all non-faulty channels.

The FCS has to process data from the various voter/monitors so as to reduce the number of different warnings to be displayed to the pilot as far as practical. Only failures which affect handling or require pilot action will be reported to the pilot and circumstantial data recorded by the system. FCS warnings are divided into two groups:

- o First and second failure of similar functions
- o Failures requiring some particular pilot actions or the observation of a restricted flight envelope.

Various surveys [4, 5, 6, 7] of fault-tolerant computing introduce many of the concepts and definitions relevant to redundant digital systems, but they need to be interpreted in the light of the FCS application. Moreover the references cited above do not all agree on the definitions. A consistent set of fault tolerance terminology should be adopted within the industry as currently different definitions are used by the customer, airframe manufacturer and equipment supplier alike.

Redundancy management strategies are presently almost exclusively directed at protection against random hardware faults and fault avoidance techniques have been the main method to achieve high integrity software. The research community has developed several approaches to the implementation of software fault tolerance, the proposals that have received the most attention are N-version programming and recovery blocks. These methods are faced with several practical difficulties in its implementation [16, 17] and will not be discussed here. Hardware diversity ("dissimilar processors") and/or analogue/digital back-up system(s) with at least level 3 handling characteristics as a concept to circumvent the existence of design (generic) faults/errors places an additional burden on the development process where the gains can't be quantified since it is impossible to predict the probability of occurrence of generic errors and this approach can create more difficulties than it removes. The inclusion of a back up system is often based on emotional feelings and/or because the purchaser does not believe that the integrity of the software can be adequately demonstrated but can lead to a relaxation of design discipline.

State-of-the-art FCS's exhibit the common characteristic of parallel redundancy in the computing section as the basic element of failure tolerance. It is a traditional issue to discuss the required number of redundant lanes whereby the discussion is limited to triplex versus quadruplex. The discussion is often

biased by the hidden implication that a triplex structural system is more advanced than a quadruplex one.

To ensure the functioning of the safety critical path as defined in section 2.3 is the prime objective and related to this set of elements, the second failure requirement determines the level of redundancy.

- o A coverage factor (a quantitative description of success in failure detection and isolation) of $c = 1$ defines a quadruplex configuration.
- o A coverage factor of $c < 1$ offers the chance to trade-off triplex versus quadruplex.

This trade-off requires the careful considerations of various aspects:

- o A triplex-redundant FCS, where emphasis is placed on self-monitoring, still requires considerable development to achieve the high coverage for the two fail operational requirement and to lower the extreme cost of verification and validation. Selfchecking processor pairs claim 100 % coverage, but double hardware complexity. The advent of VLSI offers an opportunity to reexamine existing redundancy techniques and apply them to develop FCS's that achieve not only the stringent flight safety requirements but are also more cost effective.
- o The achievable benefits from reducing a quadruplex architecture to a triplex one of the same functional range. The improvements in system weight, mission reliability and FCS defect rate have been estimated for a system as described in section 3 to lie in the range of 10 % to 15 % of these for a quadruplex system.

A triplex architecture clearly increases the complexity and hence the effort for validation of the safety critical functional path. MBB decided to adopt the less complex quadruplex system and to increase complexity and validation effort in areas where mission probability can be improved.

The FCS example under discussion is based on a quadruplex, minor frame synchronized, architecture with comparison monitoring (cross lane monitoring) as the principal method of failure detection and isolation. Self tests (in lane monitoring) are not the primary means of defence but are used to enhance the failure detection coverage in areas where otherwise defects might remain dormant in flight and to enhance the availability of secondary facilities (sensors or actuators) so as to improve overall system reliability.

The method of analytic redundancy can be used for either in-lane monitoring or to generate additional inputs to crosslane monitors. By this is meant the use of relationship among dissimilar sensors to achieve fault detection and isolation. The use of analytically generated signals has seen much research in recent years aimed at reducing the cost of redundant hardware and improving aircraft survivability by allowing more dispersion of components. Analytical redundancy will only be accepted if the analytical model can be determined with sufficient accuracy (e.g. adequate knowledge of the sensor characteristics during a failure is available and correctly modelled) prior to first flight, and therefore no additional flight test results shall be required to determine the model or any parameter associated with analytical redundancy.

3. FLIGHT CONTROL COMPUTING DESIGN CONSIDERATIONS

The general FCS design objectives defined for a fighter aircraft currently under development in Europe are flight safety, performance, reliability, maintainabil-

ity and expandability. These objectives have to be achieved by the system design while remaining within the following design constraints:

- o system mass, space and power
- o cost
 - the primary aim is to minimize life cycle cost over an in-service life of 25 years / 6000 flying hours per aircraft
- o timescales
- o operational environment
 - environmental conditions (thermal range, vibration, shock)
 - electromagnetic compatibility, lightning protection, NEMP, nuclear hardening

Reliability and maintainability (including testability) are to be given equal priority to flight safety, performance and timescales during all phases of a program.

Utilization of low-power devices, advanced VLSI technology, custom monolithic analogue microcircuit technology (CMAM), ASIC, surface mount technology together with innovative approaches to aircraft installation design and methods are major prerequisites to incorporate the required functionality in the given constraints (such as mass, space and power consumption) of the FCS. Software development, verification, validation, and maintenance is becoming an ever increasing part of aircraft life cycle cost. Advances in microelectronics, the highest possible degree of standardization, a consistent system development methodology and supporting tools (for requirement specification, design, implementation, maintenance) dealing with the entire life cycle, of which development is only a part, can collectively lower aerospace electronic system life cycle cost.

FCS computing design should anticipate needs for longterm evolvability of both the system and the system concept. As hardware costs continue to shrink relative to software costs, the reusability and portability of the system concept and the software is going to be increasingly important. System development should be based not merely on a loose collection of good ideas, techniques and tools but also on a pervasive methodology approaching emerging technologies and new requirements for improvement of weapon system performance in a system oriented manner.

FCS should be kept as simple as possible. Safety unquestionably suffers whenever unnecessary complexity creeps in and is a misguided belief in complexity as a way to achieve performance. The most difficult part of fault-tolerant FCS design is mastering the complexity. The trend towards distributed processing with "smart" sensors/actuators offers promise because it decomposes the overall complex system into smaller, comprehensible subsystems with simpler design problems. Since the functions for flight controls are naturally partitioned, the system can easily be configured into a distributed arrangement with fixed assignments of tasks. A hierarchical decentralized controls methodology reduces the complexity of the control by distributing the control authority to local controllers. This allows the design to be comprehensible at the global level while remaining optimal at the local controller- or subsystem level. This concept is already the baseline for an A/C currently under development, where an IMU (analytical platform) is supervised and controlled by a FCC and actuator loops are digitally closed by dedicated processor(s) within the (multi-processor) FCC. When "smart" actuators become available due to advances in high temperature electronics will

this not cause a change of concept but only a further decomposition of system complexity. Additional merits of this approach are enhanced fault tolerance and opportunities for improving the performance/cost for the life-cycle of the system.

Designed to a set of stringent safety requirements the FCS must be free of any architectural constraints, provide the capability to accommodate advanced hardware element retrofits to expand functions and add subsystems with minimum impact on the system.

FCS architectures for fighter aircraft in service are characterized by non-standard hardware and dedicated hardwired interfaces which impose significant weight, volume and power penalties, and allow little freedom for growth. The general way of intra-lane data exchange within the FCS shall be realized by four simplex standard, time multiplexed data buses, endowing the FCS with a high degree of flexibility combined with limited wiring expense. The selection criteria for STANAG 3838 is based on reliability, cost, industry acceptance and bandwidth. The address limitation of 31 remote terminals will not be a problem for FCS and replacing the 3838 bus by a fibre optic equivalent is feasible.

The analysis of FCS architectures for optimal functional partitioning, i.e. the definition of Line Replaceable Items (LRIs) has to minimize

- o hardware interconnections
- o signal interchange and the related timing constraints
- o failure propagation due to lack of/erroneous data
- o modifications required for future LRI enhancements and/or additions
- o workload for fault location and subsequent maintenance action.

A well designed computing system configuration should keep the complexity of system interconnections to a minimum with the constraints set by data links, LRI size and fault tolerance requirements.

The first ingredient of reliable electronic systems is a reduced chip count. FCS are I/O intensive and the challenge is thus to minimize interface hardware and wiring mass either by LSI implementation and/or by intelligent ("smart") sensors/actuators. Although there are definite trends towards distributed processing it will probably not materialize in the near future for "smart" actuators.

FCS character of computation exhibit a large degree of regularity (with low complexity) where real time aspects and phase lags are of prime importance. A maximum practical percentage of functions should be performed by program execution rather than by analogue hardware circuits.

Parallel processing is an important consideration in the FCC, with the implied requirement that tasks are partitioned and allocated to take advantage of the inherent concurrency. Distributed and parallel processing offer the benefits of smaller, more manageable and validatable software modules which are physically as well as functionally independent.

The computing system configuration must take into account the flight safety requirement of the entire FCS (including sensors and actuators).

Reliability and not convenience of programming was the primary motivation behind the development of high level structured languages. Powerful and mature development tools and environments will improve software dependability and productivity by enabling engineers to focus on requirements and design rather than on coding.

3.1 FCS Description

The system (Fig. 9) is defined as comprising

- o Sensor equipment and digital computers
 - Air Data Transducers (ADTs)
 - stick sensor assembly, pedal position sensor, cockpit switch functions
 - Air Intake Pressure Transducers (AIPTs)
 - Inertial Measurement Units (IMUs)
 - Flight Control Computers (FCCs)
- o FCS primary and secondary actuation systems including the actuators, the control surfaces and associated parts such as hinges, bearings, etc.

and performs all the tasks of measurement, computation, primary and secondary control surface actuation necessary to carry out all manoeuvres required by the pilot or mission avionics system.

3.2 Flight Control Computer

The quadruplex FCCs are the main controlling element of the FCS and have to provide the computational capability as listed in Table 2. Table 3 summarizes the FCC hardware characteristics. Each FCC shall be identical and shall contain the necessary software and hardware to operate in any lane of the FCS. The FCC multiprocessor architecture is characterized by a collection of cooperatively autonomous computer elements which are based on the local processing principle, i.e. the memories and peripheral units are firmly allocated to a CPU and cannot be addressed by external processors. Communication shall be achieved by either

- a) a high speed parallel Multi-Master-Systembus
- b) or by other methods provided that the FCC architecture does not restrict data flow and is inherently expandable.

Task distribution between computer elements has to be partitioned such that they will be almost independent of each other as far as function is concerned reducing communication between the tasks as much as possible and minimizing additional transport delay (due to communication) and real-time and hardware overhead to support concurrent execution. Static assignment of tasks to processors is required and I/O processing, loop closure of the auto-throttle actuator(s), primary and secondary actuators, and redundancy management have to be separated from control law processing and air data computation to provide flexibility for changing the control laws during life cycle and allow the later introduction of additional autopilot modes with minimum disturbance to the core program. Task scheduling is not interrupt driven by randomly timed events and interrupt sources are reduced to an absolute minimum and only allowed for

- a) triggering minor frames and/or shorter subframes
- b) exception handling.

The general method for controlling the executive sequence is by use of a time synchronized (cyclic) executive. The sequence of activities to be performed by a processor within a frame (or shorter subframe) is predetermined due to the implementation of a static preplanned (nonpreemptive) scheduling mechanism. The resulting system, while relatively easy to implement and to validate, is exceedingly inflexible once iteration rates and task selections are made. Because of the performance (not effective CPU utilization) and flexibility limits of the nonpreemptive scheduling mechanism the alternative of using priority based (pre-

emptive) periodic scheduling has to be investigated and how a proof of correctness of the cooperating concurrent processes can be performed.

To reduce development and support cost, a single processor type for all aircraft systems is highly desirable. In order not to constrain hardware design by equipment suppliers unnecessarily, component standardization should be limited to the microprocessor. Wherever processor type standardization is required, the decision on the particular type of component to be used will require protection for supply and support of the product for the life of the aircraft or an equipment upgrade path to be identified which preserves investment in system/SW design. Comparing the suitability of different processors is problematic because many variables (including commercial conditions) are involved and even the criteria by which they are judged are controversial [8, 9, 10, 11].

Control functions to be performed in the FCC in the future are becoming more comprehensive and complicated as task oriented control and carefree manoeuvring modes are fully exploited. The multiprocessor configuration ensures that sufficient processing throughput is available as well as allows the control tasks to be partitioned into convenient modules.

Frequently used software functions have a tendency to migrate into hardware - possibly via firmware as an intermediate step. Thus some of today's software issues can be expected to continue to become hardware issues in the future. The software features that are particular strong candidates for hardware realization in the future include voting/monitoring, distributed control features for synchronization, intercommunication and scheduling. Scope control and data encapsulation - essential in modern programming languages like Ada - can be provided in hardware with better addressing mechanisms that integrate advanced approaches to protection, eliminating the so called "semantic gap" of the von Neumann machine [12]. Objects and capabilities, along with modularity, will contribute to reliability due to a fine grain of protection and fault containment.

Recently, considerable attention has been focused on RISC machines motivated primarily by its low hardware requirements and claimed higher throughput. As stated in [12]: In terms of protection, however, RISC machines are a step back to the dark ages of the "pure" von Neumann machines. This issue has recently been discussed by D. Nelson who characterises the RISC machine by using Ralph Nader's famous slogan "unsafe at any speed".

General-purpose, single-chip (monolithic) digital signal processing (DSP) devices are providing new and expanded applications in the area of LVDT/RVDT signal conditioning and digital control loop closure of actuators.

3.3 Sensor Interface

The sensor interface covers the sensors necessary to meet the airframe related requirements as well as those to meet the mission related requirements. Below, only the former is outlined. The interface to the latter is via the AVS and UCS bus system. The analysis of the functional requirements in the sensor area revealed a commonality of the FCS and other A/C systems for air data and inertial data. Both low accuracy with high availability and high accuracy with low availability sensor data are needed in the FCS for guidance and control functions. The requirements of the AVS (e.g. Displays, Navigation, Weapon System) and the UCS (e.g. Engine Control Fuel Gauging, Warnings) are covered by those of the FCS. As the objective was to optimize overall installation, hardware required for the implementation of one system had to be utilized for implementing other functions. This led to a centralized ADS. Apart from the above mentioned advantages with respect to installation, there are a number of other advantages (e.g. Location error correction). Structural effects made it necessary to measure inertial data for the AVS and the FCS at different locations in the A/C.

Further, a central IMS would not have been weight and cost optimal. Therefore, a high accuracy/low availability system being part of the AVS and a medium accuracy/high availability being part of the FCS was chosen. The latter, however, is accurate enough to provide the back-up navigation function.

The ADS is based on a four multi-function probe configuration (skewed sensor arrangement) to give independent measurement of pitot and static pressure and local flow angle (see Fig. 10). Each probe is combined with pressure transducers and 16 bit microprocessor based electronics for computation and BIT to form an Air Data Transducer Unit (ADT) avoiding pneumatic pipework and water separators. The ADT measures and transmits local flow angle (ϕ) and indicated static/true pitot pressure to its own lane FCC via a simplex STANAG 3838 FCS bus where all air data compensation, computing, voting and monitoring will be performed. Probe heater control and monitoring is performed by the ADT.

Each ADT is supplied with semi-conditioned power from the appropriate FCCs to minimize the size of the sensor unit.

The FCS inertial measurement system consist of four attitude and heading strapped down platforms. The sensed A/C rates and accelerations and various calculated parameters are used within the FCS to provide artificial stabilisation of the A/C and are also transmitted by the FCS to other systems e.g. to support backup navigation. The FCS inertial measurement system forms an integral part of the aircraft inertial measurement system.

Each IMU lane consists of gyros, accelerometers and associated electronics for signal conditioning, computation, data transmission and BIT implementing a strapped down attitude and heading reference system by measuring linear acceleration and angular rates in all three body axis. From these, the IMUs calculate and output to the FCCs on a lane by lane basis via simplex STANAG 3838 FCS buses the required signals (see Fig. 11). With a skewed sensor arrangement the number of sensors and equipment mass can be minimized.

Cockpit and switch functions (with the exception of Multi Head Down Display soft key functions and autopilot selections) are hardwired to the appropriate FCCs, and they provide the necessary excitation for each sensor/transducer. The considerable number of FCS signals associated with the cockpit causes the airframe designer problems to route the cables through the cockpit bulkheads and incurs wiring mass. A four lane cockpit interface unit (CIFU) which receives pre-conditioned power from the FCCs, and interfaces with the appropriate STANAG 3838 FCS bus can incorporate the discrete inputs/outputs and analogue input functions. The integrity implications and the resultant effect on mass have to be investigated.

3.4 Actuator Interface Configuration

Direct drive valves (DDVs) represent the actuator system related technology to be specified for fighter A/C currently under development. DDV benefits occur primarily because of the simplified interface of multiple digital signal processing channels with two hydraulic systems (i.e. effects of electrical and hydraulic failures are separated), reduced electrical wire count, substantial weight savings, the inherent simplicity and reliability and hydraulic system savings by eliminating EHV null flow losses.

The simplest actuator design utilises a single stage (linear or rotary) DDV (see Fig. 12). Each FCC drives one lane of the direct drive motor, the mechanical output of which is directly connected to a high precision hydraulic control valve. The output of this control valve is connected to a tandem hydraulic ram, thus controlling its position. The position of the hydraulic

control valve is measured by four LVDTs, the output of each being fed back to a different FCC to close the control valve position loop. Quadruplex voted and monitored ram LVDT feedback signals are used to close the main ram position loop. Loop closure of the inner and outer loop will be digital with a sampling frequency of 320 Hz and 80 Hz respectively. A jammed valve must be detected, and the motor drive amplifier has to be capable of providing enough current to overcome the valve jamming force. Due to the low probability of a jammed valve occurring, the capability to overcome the jam force is only required with all four electrical control lanes operating at normal aircraft power supply voltage and both hydraulic systems functioning. The probability of loss of control valve function due to a hardover failure in one lane, which cannot be switched off, followed by a subsequent failure in another FCC shall be reduced to less than 10^{-9} /FH.

Digital control loop closure provides means of reducing inner loop error signal disparity between lanes in a high gain actuation system by low rate equalization of control valve LVDTs, thus reducing force fight in the DDV and gives scope for reduction in matching requirements (hence cost) of the position transducer.

Problem areas of potential concern are increased electrical power levels, EMI of power switching electronics and magnetic coupling in the DDV. Pulse width modulation allow the valve driver amplifier to act as a switch rather than as linear (class A) amplifier, thereby improving the amplifier efficiency and reliability as a result of the lower power dissipation.

The specification of actuator interfaces causes the airframe manufacturer problems, because actuator characteristics and its controlling FCC influence each other's design requirements. Cross-company definition and responsibility problems are to be solved and therefore the FCC Supplier and the actuator Supplier have to cooperate to achieve the best performance of the actuation systems. The FCC Supplier has to provide sufficient information about actuator loop closure techniques, to enable the actuator Supplier to design a test set with representative simulation of the actuator drive, feedback demodulator, digital feedback paths as provided by the FCC actuator interface, so that representative actuator performance testing can be executed.

"Smart" actuators, i.e. with actuator mounted, or local, electronics for actuator loop closure, redundancy management and BIT are a technology goal to link actuators directly to the FCS buses with the following advantages:

- o easier to specify for airframe manufacturers
- o actuation system can be purchased from one Supplier
- o system integration via FCS bus, e.g. electronic or optical
- o low weight system
- o improved reliability
- o reduced system vulnerability to lightning, EMP, RFI
- o reduced demand for fibre optic sensors

High temperature electronics for "Smart" actuators have to survive in an uncontrolled environment. GaAs is a semiconductor technology for environmental extremes [13], and its development is supported by DARPA and SDI because of its vital military applications (potential for high-speed signal processing, low power dissipation, high radiation resistance).

The use of optical encoders to measure linear and angular displacement is limited by timescale and connector density. Fibre optic sensors have seen much research in recent years aimed at reducing system vulnerability to lightning, EMP and RFI, but this technology is not mature enough to be used for A/C under development.

3.5 Software

The necessity of using tools for system analysis, software requirement analysis, software development and maintenance, which allow requirements to be analyzed for completeness and consistency and provide support for development phase independent activities (e.g. configuration management, project management, quality assurance, documentation), is no longer disputed even in the practical domain. An Integrated Project Support Environment (IPSE) has to be built up, comprising a set of tools, dealing with the entire life cycle. The IPSE will provide integration of the tools on two levels, first at the user level and second at the data level. The integration at the user level will provide a common user interface to the tools, which have been fully integrated with the IPSE. The integration at the data level will allow different tools to access the same data and allow the relationship between the outputs from different phases of the software life-cycle to be recorded.

It is essential that the FCS is not designed in isolation from other systems, thus each system has to follow a common design method where the functional and performance requirements are conducted in a phased manner following a design route. All output documents produced at the end of a phase have to be reviewed and validated as an overall A/C system activity.

System analysis is divided into three phases. Phase 1 specifies FCS top level requirements and interface requirements to other A/C systems. Phase 2 will produce the FCS functional requirements and the Preliminary Interface Control Documents. Phase 3 will address the individual LRIs and will produce for these the LRI Processing Specification and the Interface Control Document (ICD). The output documents of phase 3 will be given to the selected equipment supplier. The management, development and production of the procured software will be in accordance with software standards and software management control procedures (based on DOD-STD-2167) specified by the airframe manufacturer. The equipment Supplier starts with a software requirement analysis phase and produces a Software Requirement Specification. A detailed description of the software phases lies beyond the scope of this article. The paper by Cavano [15] discusses from a DOD perspective a management approach to achieving high confidence software, highlighting software reliability as the key factor and emphasizes that the software reliability must be continuously monitored over its life cycle.

The integrity of the FCS software has to be compatible with the given integrity requirement for the system and is regarded as class 1 (safety critical). Currently Assembler is the language used most for FCS software implementation where safety and performance are the main conflicting requirements. By comparison, much software outside the safety critical area has long been written in High Order Languages (Jovial, Pascal, Ada). Ada directly supports such vital software engineering concepts as data abstraction, information hiding and modular design. The advantages over Assembler are well known and need not to be expanded upon here. To take advantage offered by Ada, a careful assessment was undertaken to prove whether Ada can be used to produce an implementation of equivalent integrity to an Assembler one. To obtain an answer to this question three major problem areas had to be addressed

- o Is Ada able to solve safety critical problems
- o How could the Ada-Compiler integrity be assessed
- o What steps and procedures must apply in an Ada-implementation for verification and certification purposes.

To answer question one a definition of safety had to be defined which was consistent with the accepted software engineering practice for current FCS implementation and second, based on this definition the language features defined in the Ada Language Reference Manual were examined with regard to

whether or not their use would have an impact on this safety definition. This examination resulted in a list of 35 language restrictions to be imposed when applying Ada to safety critical areas. The obedience of these restrictions can be checked for each program, maybe by automatic tools.

The Ada-Compiler integrity can be assessed by verifying the target code to any demanded degree with the means of Code Mapping Charts to provide full visibility of the compilation process.

The use of code analysis techniques using static analysis tools allows verification that the Ada design and code implements the functional and safety specification of the initial design. Manual compilation of target code by use of Ada to target code mapping charts demonstrates the visibility and correctness of the source to target code mapping performed by the compilation process and thus gives confidence in the compiler generated target code and in the compiler itself.

The processing overhead to be paid for Ada generated code is still in debate. Efficient floating-point Ada target code generation is important for FCS software in order to eliminate the need for fixed point arithmetic. The relatively high number of fast floating-point registers in modern floating-point co-processors makes register optimization much more important. The compiler must schedule register use over the entire course of a program, paying attention to how often the value appears in the source code as well as how often it will be used during execution.

The software of less complex LRIs (e.g. ADT) may be written in Assembler.

3.6 Operational Environment

The environmental conditions experienced by each LRI are determined by its location in the aircraft. Means to fulfill the structural design requirements for vibration, gun fire vibration and shock are well known and need not to be expanded upon here. The ability to operate in absence of a cooling medium for at least 15 minutes is required and this event has to be indicated to the pilot. The FCS equipment and their installation have to be protected against lightning strikes and must be electromagnetically compatible with other A/C equipment ("internal compatibility") as well as be capable to operate in a substantial external radio frequency/electromagnetic environment. A production testing inspection plan has to demonstrate that the EMC properties will be maintained during equipment production.

Design to protect against the NEMP threat (EXO- or ENDO NEMP) is basically the same as designing against EMI (external CW RF fields) or lightning protection. The biggest threat to the FCS computing system is posed by Initial Nuclear Radiation (INR). Shielding against these penetrating rays is completely out of question because approximately 5 cm lead would be required to reduce the prompt gamma level by a factor of ten. In general, nuclear hardening techniques are divided into two classes:

- o Intrinsic Hardening depends mainly on the exclusive selection of components which are proven to meet a project radiation requirement. This will result in a FCS which provides uninterrupted performance in the environment specified.
- o Extrinsic Hardening depends on the use of only a minor number of "radiation hard" components, i.e. the system is protected by utilizing architectural means. The FCS has to detect the nuclear event, must take corrective measures against latch up of semiconductor circuits and has to

return to full performance (or at least State 3) after a specified time interval.

The time to double amplitude (a measure of how rapidly a dynamic system diverges) of a CCV fighter aircraft, which is inherently unstable over a large part of the flight envelope, is in the order of 0.2 sec. The majority of discrete components and semiconductor circuits are not designed with any INR requirement in mind, even in military applications, and information on INR hardness of available Integrated Circuits is extremely limited. Because of the US SDI programme it can be assumed that within 5 to 10 years INR hardened components will become widely available.

3.7 Testability and Maintainability Issues

The FCS BIT objective is to detect failures either as they arise during flight or prior to flight during the preflight check, and to ensure that the system is at all times operating within defined performance limits. Because redundant systems lend themselves to BIT by their very nature, a concept of maximum coverage is to be developed with not only flight operations, but also test and maintenance activities considered from the start. According to the operational mode of the aircraft the FCS will be tested by:

- o Continuous Built-In-Test (CBIT)
- o or Initiated Built-In-Test (IBIT)

IBIT will be divided into three operational levels:

- o Pre-Flight Check (PFC) will be initiated automatically each time the FCS is powered up on ground and comprises purely electronic checks. The actuators will be maintained in safe position.
- o Actuator Checks will be carried out on a weekly basis (12 op hr) and are conditional on successful completion of the PFC and hydraulic pressure being available.
- o First Line Checks will be used to investigate noncritical areas of the FCS (for diagnostic reasons), flight critical (at 600 hours period of risk) sections, which cannot or need not be checked in PFC, and also to operate interactively with other aircraft systems, ground crew etc. where this is impractical within the constraints of PFC.

The three IBIT functions are to be designed in an integrated manner but will be executed in accordance with the FCS maintenance/servicing concept. However, each level may serve as a maintenance test depending on the nature of maintenance action required.

FCS CBIT provides failure detection, isolation and reporting by a combination of Cross-Lane monitoring and In-Lane monitoring. Trend monitoring (preventative maintenance) can be achieved by monitoring the number of occasions that the individual voter/monitor updown counters operate, to determine those signal failure which are transient in nature and thus may fail positively in the future.

The FCS is part of a virtual integrated on-board test, evaluation and recording system, which will be distributed across all aircraft electrically/electronically monitored systems in order to minimize false alarms, maximize fault isolation and identify intermittent failures. BIT failure information gained from both, CBIT and IBIT, will be reported as it occurs to an Integrated Monitoring and Recording System. In addition this information is also retained in the LRIs on non-volatile memory and can be accessed via the data bus or test connector at flight line or shop level.

The FCS has to provide the necessary test interfaces for support equipment to give test assistance throughout a project life cycle, i.e. FCS development, integration, aircraft integration, structural coupling, EMC and production aircraft testing.

Modularity and common standards improve maintainability and reduce acquisition costs by simplifying the development of off-the-shelf items that can be integrated into future systems. The US Air Force has established the Pave Pillar avionics architecture [14] that allows a two-level maintenance concept to be implemented. In today's standard three-level maintenance concept (flight line, intermediate shop, depot/factory) with removal/ replacement of LRIs at the flight line the need for the intermediate shop level can be excluded when modules become Line Replaceable Modules (LRM). The merits of the LRM concept with a semi permanent installed integration rack include A/C installation improvements, reduced weight and EMI susceptibility, and ultimately a reduction in spare inventories. No attempt is made here to discuss the life cycle cost aspects - these are more than adequately covered in the Pave Pillar article in this special issue.

Artificial intelligence techniques (Expert Systems) [19] can be used in future for an integrated diagnostic application.

4. RESTRICTIONS

Flight Control Systems must meet stringent specifications of flight safety and availability. Conservative design practice in engineering imposes certain restrictions upon the application of advanced computer hardware/software and micro-electronic technologies. The discussion centers around two specific items:

- o VLSI testability and configuration control
- o Software reliability

While VLSI offers numerous opportunities, it also introduces a number of new problems for FCS, which have to be considered. The clearance of industry standard (commercial) microprocessors and other complex electronic circuits give rise to the following concerns:

- o Commercial definition specification, which is often ambiguous
- o Difference between hardware and documentation
- o Subject to continuing development, e.g. mask sets are updated, without the manufacturer informing the user.
- o Changes added by second source manufacturer
- o Transient faults will be more prevalent because extremely low circuit energy level will make devices much more susceptible to external interferences
- o Too complex for totally rigorous test in all possible operating conditions

Theoretically, neither simulation nor testing can completely verify the correctness of a VLSI circuit. The acceptance problems are further increased due to a lack of a formal mathematical specification for complex instruction set computers (CISC). As VLSI designs increase, the issue of correctness grows from an interesting theoretical question into a practical consideration with which equipment Supplier and airframe manufacturer must deal. One cannot rule out the possibility of residual hardware design faults at the chip level. Of grave con-

cern is the possibility that one or more very rarely entered processor states might represent a hazardous generic design flow and that such a state might be entered into (essentially simultaneously) by all processors in all redundant lanes [7]. The same problem can show up with ASICs, where the equipment Supplier is responsible for the testing procedure and no high-volume customers can assist in finding errors. This fact reinforces the importance of using industry-standard processors.

Methods of alleviating type acceptance problems are strict configuration control to microprocessor- and other VLSI circuit mask level, traceability of each batch of components used and robust design of computer architecture to detect/absorb unexpected VLSI behaviour. MIL-STD 883C production does not qualify one standard of mask set, although CECC 90000 does.

The absence of credible reliability prediction methods for safety critical (class 1) software makes the FCS reliability analysis questionable. The length of time under test and the small samples of failure observations make it impractical to assess the reliability of the software by means of growth models and statistical models [18].

FCS software is only modestly complicated. The key principles which FCS software must be based on are simplicity and visibility. These principles must be manifested within the specification and production process as well as within the products (reports) generated during the different development stages. The term "simplicity" does not mean, that the overall system complexity must not proceed some simple level, but it does mean that on each specific stage the relevant information can be simply surveyed and reviewed by human analysts. In order to achieve the required quality standard an intensive control of the software production procedures is indispensable including clear and thorough definition of requirements, extensive testing and design audits, detailed documentation, and rigorous production and configuration control.

In practice of course no matter how carefully the software is designed it is impossible to establish that it is completely error free since the large number of possible states preclude exhaustive testing and the statistical analysis methods used in hardware design are not applicable to software. Since we know of no satisfactory way to estimate the probability that a software module is incorrect, we are forced to try to guarantee that the software is indeed correct.

A "Proof-of-Correctness" method [20] for validating software is being investigated by SRI Inc. as part of the SIFT Program. This methodology employs a hierarchy of design specifications from very abstract description of system function down to actual implementation, that can be proved via the techniques of mathematical logic (predicate calculus). The most abstract design specification can be used to verify that the system functions correctly and with the desired reliability. A succession of low-level models refine these specifications to the actual implementation, and can be used to demonstrate that the implementation has the properties claimed at the abstract design specification. It must be noted that the SIFT proof of correctness only covers the operation system and not the application domain. This technique is also not readily understandable to the engineering community and formal specification with proof of correctness for real flight control systems may still be years off.

In practice very small and simple software modules must be used which are easy to verify and it is hoped that by such techniques the required high integrity can be achieved.

Current trends will increase the problem because VLSI circuits are just as complex as software.

5. FUTURE FLIGHT CONTROL SYSTEMS

A scenario which is characterized by

- o a threat due to an enemy which outnumbers the own forces and increasingly improves the performance of his aircraft,
- o the need to minimize the cost of ownership

requires future weapon systems to meet stringent requirements which extend the field of classical flight control to the field of flight guidance.

5.1 Weapon System Requirements

The need for a better survivability drives future aircraft to operate at even lower altitudes for a longer part of the mission than today's and to avoid the usage of active (i. e. emitting) sensors. The requirement to improve weapon system effectiveness involves day/night and all weather operation, the capability of the weapon system to fight several targets at the same time, to operate jointly with other aircraft and ground stations, and to maximize the mission success rate. An optimal weapon system performance can only be achieved if the pilot is relieved of tasks which are better performed by the machine. The goal is to allow him to concentrate on tactical decision making. Further weapon system requirements are an enhancement of the operational availability and a reduction in life cycle costs by improving reliability, maintainability and testability. Finally, the number of aborted missions due to system failures has to be minimized.

The resulting requirements for the computing configuration of a future guidance and control system are

- o calculation of a mission optimal demanded flight path taking into account the performance of all the subsystems involved
- o contribution to an optimized performance of the air vehicle including airframe, flight control system, engine and other airframe related systems
- o system partitioning such that each subsystem can be optimized with respect to performance, weight, volume and power consumption and maintenance without affecting other subsystems
- o functional subsystem integration such that the resulting system at a higher level is optimized with respect to performance, safety and availability

5.2 Trends in Future Guidance and Control Systems

Flight control system changes are evolutionary. While micro-electronic technology will achieve further improvements as listed below, this will not lead to FCS concepts that are radically different from those outlined in section 3. The trend to computing configurations with a distributed arrangement of elements with a fixed assignment of tasks is most likely to continue. The communication between these elements will be via standardized interfaces. This allows to specify, develop and test each element with a minimum impact on the overall system. By making the subsystems intelligent, i.e. provision with computational power performance, weight, maintainability etc. will be improved. This has partly been achieved in the sensor area but not yet for actuators due to availability of of suitable components. Although today the actuators are controlled by dedicated microprocessors located in the FCC, the above mentioned improvements are not possible due to interface restrictions. Local data processing for example example could allow the implementation of active flutter damping, thus reducing

weight and raising bandwidth. The local storage of calibration data and automatic LVDT adjustment by going through a learning process would improve maintenance and reduce life cycle cost. Also, the degree of redundancy can then be adjusted individually.

The weapon system requirements outlined in section 5.1 will lead to a functional integration of systems in two areas:

- o vehicle management
- o integration of the vehicle with the avionic sensor and mission system

The main task here is a partitioning into functional blocks and the definition of a topology so that the resulting integrated system meets the performance and availability requirements. As tasks which were previously done by the pilot are absorbed into the machine, this will result in changes to the man-machine interface. Work in the future will concentrate on optimizing the amount and type of information conveyed to the pilot and on techniques of transforming pilot's decisions into appropriate actions of the machine.

5.3 Enabling Technologies and Tools

The rapidly evolving micro-electronic technology continuously affords new opportunities in FCS concepts while at the same time posing new challenges for the effective realization of its potential. Several prospective future trends in technology are summarized here:

- o Great increase in the processing power of computer components (32 bit processors with floating-point arithmetic on chip, DSP, microprocessors optimized for artificial intelligence languages)
- o Great improvement in cost and performance among all electronic integrated circuits, surface mounted packaging, fibre-optics
- o Improved computer architectures (object-oriented)
- o Electronics for environmental extremes (temperature, INR) will be available
- o Improved aircraft installation designs (semi permanent installed integration rack assembly, LRM concept)
- o Substantial increase in bandwidth of bus systems (STANAG 3910)
- o More experience with distributed systems
- o Improved understanding of software engineering principles
- o Improved fault tolerant system design methods (including analytical redundancy)
- o Improved tools for requirement specification and software design
- o New methodologies (formal specification) for verification and validation of FCS
- o More experience with artificial intelligence techniques (expert systems for real-time control, verification and validation of expert systems)

We also have to consider what research areas not currently central to the FCS application are likely to emerge in the next five or ten years and what their

expected impact is. The functional complexity of future guidance and control systems requires the application of a strict methodology for all phases of the equipment life cycle. Today, a phased development process has been established. Tools and techniques from the enabling technologies above have to be identified and implemented which support this development process.

6. SUMMARY AND CONCLUSIONS

In the course of this paper we have first analyzed the requirements and then examined the design considerations of present and future FCS computing subsystems. The flight control computing requirements result from the weapon system carrier functions and the avionic system functions. The complexity and variety of tasks that flight control systems are called upon to perform has been steadily increasing. A/C subsystem integration and modularity will become the governing concept.

With the mentioned 35 language restrictions to be imposed Ada can be used for flight control implementation.

Nuclear hardening will become one of the biggest problems for FCS electronic subsystems to be solved.

Strict configuration control of VLSI mask sets and robust computer design can alleviate type acceptance problems. Formal specifications with proof of correctness are still years off.

Future weapon system requirements will be driven by an increased threat and the need to minimize the cost of ownership.

The trend towards distributed processing is most likely to continue because it decomposes the overall system into smaller comprehensible subsystems. System integration can be achieved by a hierarchical decentralized control methodology.

Physically, the computing system will be based on standardized LRMs.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the valuable assistance of their colleagues in the Flight Guidance and Control department of MBB in the preparation of this paper.

REFERENCES:

- [1] Computers have control, Flight International, October 5, 1985, pp 29-31
- [2] H.J. Kaul, F. Sella, M.J. Walker, 'The Flight Control System For The Experimental Aircraft Programme (EAP) Demonstrator Aircraft', Active Control Systems-Review, Evaluation and Projections, AGARD Conference Proceedings No. 384.
- [3] K.W. Rosenberg and A.R. Heaps, 'Trends In Digital Flight Control Systems', Combat Aircraft Controls Division, Marconi Avionics Ltd, Rochester, Kent.
- [4] T. Anderson and P. Lee, Fault Tolerance, Principles and Practice, Englewood Cliffs, NJ: Prentice Hall, 1981

- [5] J.C. Laprie, 'Dependable computing and fault tolerance: concepts and terminology', Proc FTCS 12, Ann Arbor, Mich, June 1985, pp 1-11.
- [6] J. Goldberg, 'Challenges and directions in fault tolerant computing', Proceedings of a Brown Boveri Symposium on Computer Systems for Process Control, September 2-3, 1985, Baden, Switzerland, pp 33-65.
- [7] L.J. Yount, 'Generic Fault-Tolerance Techniques For Critical Avionics Systems', presented at the AIAA Guidance and Control Conference, Snowmass, Colorado, June 1985
- [8] W. Suydam, 'Developers debate merits of military microprocessors', Computer Design, Feb. 1, 1987, pp 38-44.
- [9] S. Hafner, 'Can Commercial Semiconductors Meet MIL-Spec Needs?', Defence Electronics, Aug. 1987, pp 48-50.
- [10] R.N. Castellano, 'VHSIC Program Spurs U.S. IC Technology', Defence Electronics, July 1986, pp 114-127.
- [11] J.W. Rawles, 'VHSIC Technology Comes On Line', Defence Electronics, Aug. 1987, pp 42-47.
- [12] W.K. Giloi, 'Principles of computer architecture', Proceedings of a Brown Boveri Symposium on Computer Systems for Process Control, September 2-3, 1985, Baden, Switzerland, pp 9-30.
- [13] V. Milutinovic, 'GaAs Microprocessor Technology', Computer, October 1986, pp 10-13.
- [14] J.C. Ostgaard and R. Szkody, 'Pave Pillar Avionics Designed for Dependability', Defence Electronics, May 1986, pp 79-89
- [15] J.P. Cavano, 'Toward High Confidence Software', IEEE Transaction on Software Engineering, Vol SE-11, No. 12, pp 1449-1455.
- [16] N.G. Leveson, 'Software fault tolerance in safety critical applications,' Proc. 3rd International GI/ITG/GMA Conference on Fault-Tolerant Computing Systems, Bremerhaven, Germany, September 1987, pp 1-12.
- [17] D.E. Eckardt and L.D. Lee, 'A theoretical basis for the analysis of muliti-version software subject to coincident errors', IEEE Trans. on Software Engineering, Vol SE-11, No. 12, pp 1511-1517, 1985.
- [18] A.L. Goel, 'Software Reliability Models: Assumptions, Limitations, and Applicability', IEEE Transaction on Software Engineering, Vol SE-11, No. 12, pp 1411-1423.
- [19] Knowledge Based Concepts and Artificial Intelligence: Applications to Guidance and Control, AGARD Lecture Series No. 155.
- [20] P.M. Melliar-Smith, R.L. Schwartz, 'Formal Specification and Mechanical Verification of SIFT: A Fault-Tolerant Flight Control System', IEEE Transactions on Computers, Vol C-31, No. 7, July 1982, pp 616-630.

FIGURES AND TABLES

- Fig. 1: Pedigree of MBB's FCS Activities
 Fig. 2: Flight Controller Functions
 Fig. 3: Schematic to Define the Computing System of the FCS
 Fig. 4: Control System Model to Define Allowable Instability Level
 Fig. 5: Aerodynamic Control Surfaces
 Fig. 6: Flight Controller Structure
 Fig. 7: Schematic of Automated Functions
 Fig. 8: Relation between Operational States and Safety/Mission Requirements
 Fig. 9: Schematic of FCS Architecture
 Fig. 10: Schematic of the Air Data System
 Fig. 11: Schematic of the FCS Inertial Measurement System
 Fig. 12: Schematic of the DDV Actuator Control

- Table 1: Definition of Operational States
 Table 2: FCC Computational Capability
 Table 3: FCC Hardware Characteristics

Table 1: Definition of Operational States

| Operational State | Definitions (MIL-F-9490D) | Functions | Performances |
|-------------------|---|--|--|
| I | Normal operation | 1 SAS/CAS 2 Configuration control 3 Carefree manoeuvring 4 Basis autopilot 5 Automated functions | ● Full performance within operational envelope |
| II | Restricted operation | SAS/CAS Functions 2-5 partially inhibited (failed) (not design criteria) | ● Degraded mission performance ● Restricted flight envelope ● Increased pilot workload |
| III | Minimum safe operation | SAS/CAS Minimal path of safety critical function | ● Loss of mission performance ● High pilot workload - safe for landing |
| IV | Controllable to an evacuable flight condition | SAS/CAS Functions time limited (sec!) | ● Satisfactory for ejection |
| V | Catastrophic condition | Immediate loss of controllability | ---- |

Table 2: Flight Control Computer Computational Capabilities

The FCC has to provide the following computational capabilities:

- o the interlane communication required to implement the FCS redundancy management
- o to control BIT functions at the level of inflight, preflight and first line tests to verify that the FCS is capable of meeting satisfactory performance requirements and fault isolation
- o to implement the mode logic and the control laws required by the primary functions and secondary actuation systems
- o for the air data processing
- o the BC interface to control the serial digital intralane communication (simplex STANAG 3838 bus)
- o the interfaces to implement the primary and secondary actuator control loops and the interface with the auto-throttle actuator(s)
- o the required analogue and discrete I/O interfaces for the FCS related cockpit functions and the capability to control the proper sequencing
- o to service the STANAG 3838/3910 RT to the Avionics and Utilities buses
- o to provide preconditioned power for ADT
- o to service for accident/incident data recording
- o the interfaces to support equipment (e.g. Flight Test Data Acquisition Unit for the development and flight test activities)
- o a software controlled frequency signal generator for investigation of flying qualities (frequency response measurement, identification of aerodynamic stability and control parameters)

Table 3: Flight Control Computer Hardware Characteristics

| | |
|-----------------------|--|
| Computer Architecture | Multi-processor system |
| CPU | = 4 industry standard 32 bit microprocessors with floating point co-processors |
| EPROM | > 320 K bytes |
| RAM | > 64 K bytes |
| Nonvolatile RAM | 4 K bytes |
| Interface | Cross Channel Data Link (CCDL) with 0.8 x 10exp.5 data words/sec simplex STANAG 3838 BC dual redundant standby STANAG 3838/3910 RT RS 422 Test Interface 50 LVDT/RVDT Signal Conditions 14 DDV/EHSV Amplifiers 70 discrete I/Os comprehensive BIT |
| Growth potential | 50 % spare memory 30 % spare I/O 50 % throughput for control law, air data etc. 30 % throughput for I/O processing |
| Environmental | Ability to operate in absence of cooling medium for > 15 minutes |
| Power | 120 W (excluding power supplied to external devices) |
| MTBF | > 2500 hours |
| Packaging | 1/2 - 3/4 ATR short |
| Weight | 10 - 11 kg |

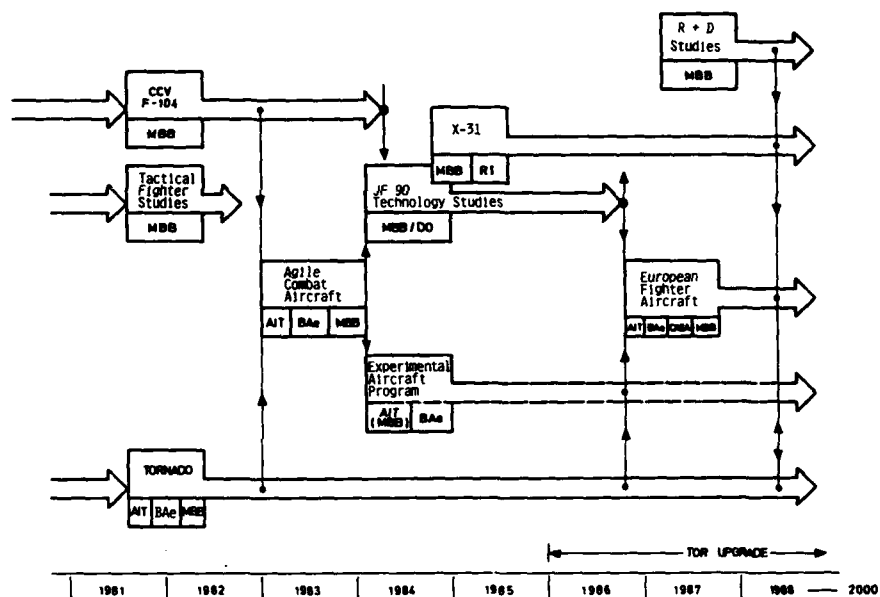


Fig. 1: Pedigree of MBB's FCS Activities

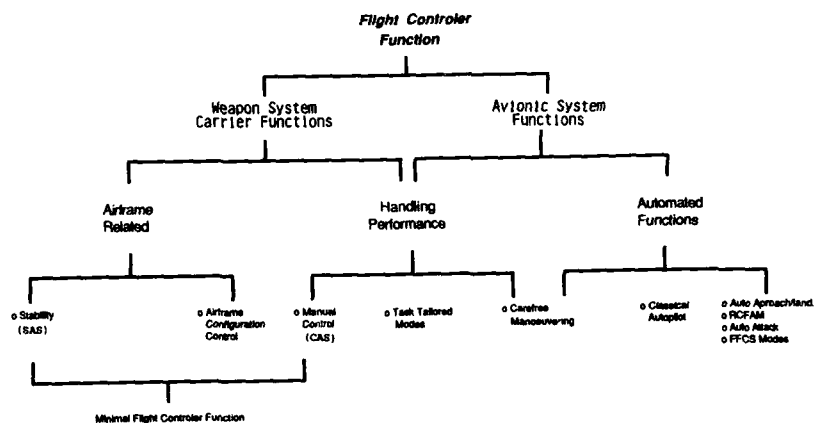


Fig. 2: Flight Controller Functions

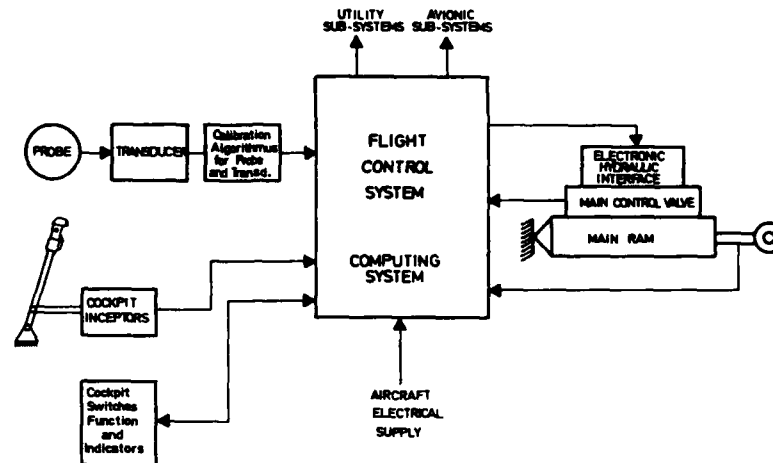


Fig. 3: SCHEMATIC TO DEFINE THE COMPUTING SYSTEM OF THE FCS

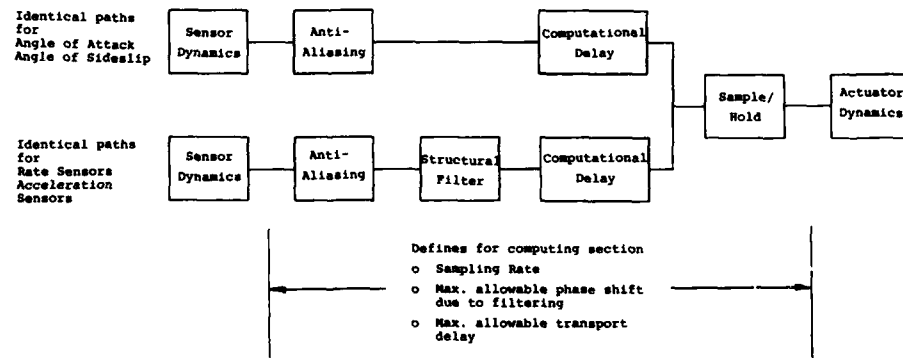
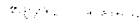


Fig. 4: Control System Model to define allowable instability level



^a $\chi^2 = 1.0$, $df = 1$, $p = .32$.



^a $\chi^2 = 1.0$, $df = 1$, $p = .32$.

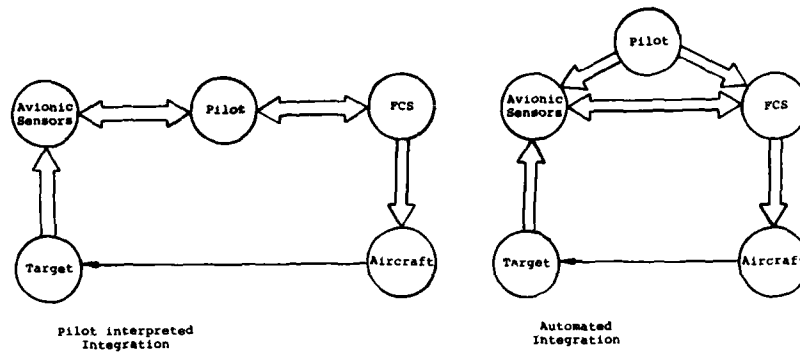


Fig. 7: Schematic of Automated Functions

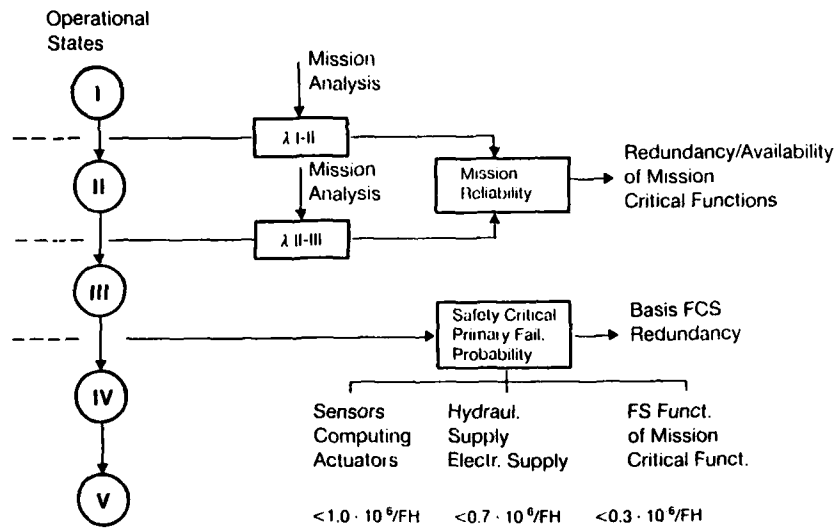


Fig. 8: Relation between Operational States and safety/mission requirements

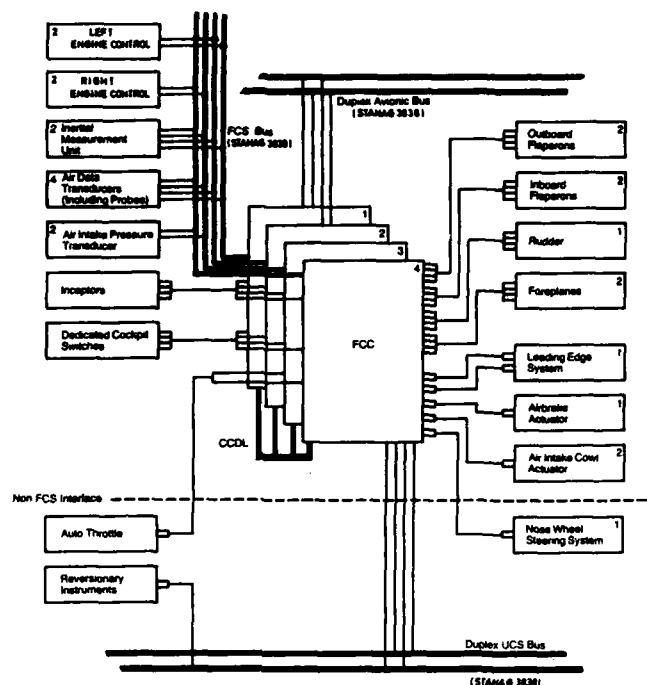


Fig. 9: Schematic of FCS Architecture

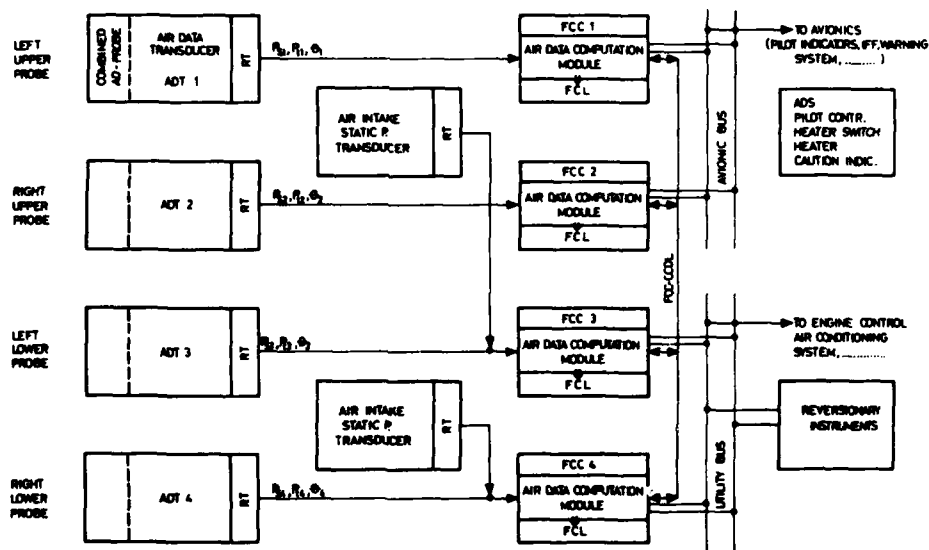


FIG. 10: SCHEMATIC OF THE AIR DATA SYSTEM

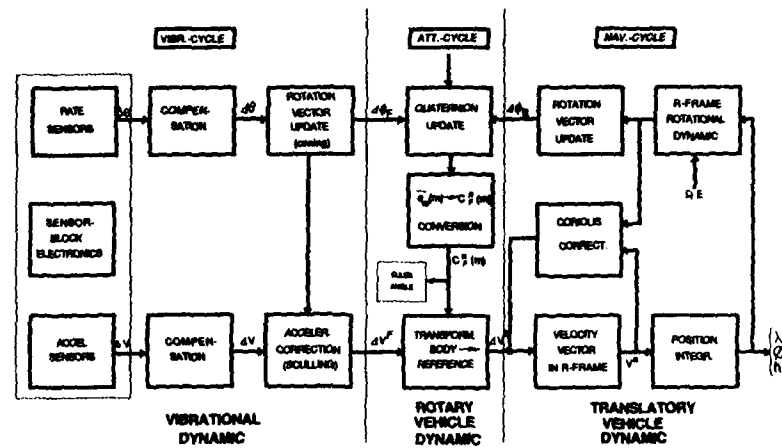


Fig. 11: Schematic of the FCS Inertial Measurement System

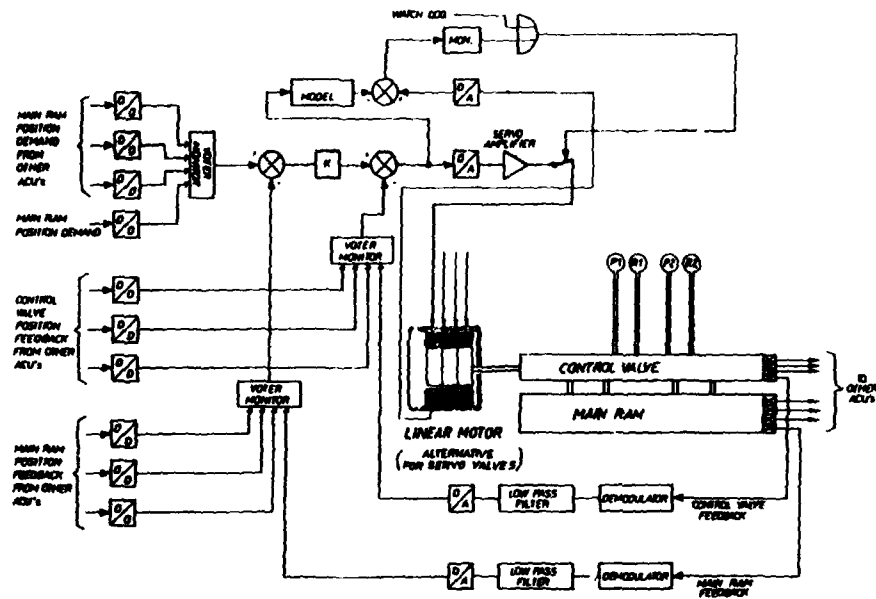


Fig. 12: Schematic of the DDV Actuator Control

A HIGHLY RELIABLE COMPUTER FOR AIRBORNE APPLICATIONS

by

Michel Muenier

Electronique Serge DASSAULT

55, quai Marcel Dassault - 92214 Saint-Cloud - FRANCE

SUMMARY

The purpose of this paper is to highlight the reliability features of the CMF⁽¹⁾ AIR embedded computer supporting its operational software. This computer was designed to meet the requirements of the ACE⁽²⁾/RAFALE D aircraft. For this reason, it has to be highly dependable, to satisfy severe physical requirements (size, weight, power consumption, etc.) and to support real-time software execution (this software being written in high-level languages such as Ada⁽³⁾ and LTR3 (a French Pascal-like real-time language)).

The general structure of the computer is first presented, emphasis being placed on the technological choices (for example, the use of ASICs : Application Specific Integrated Circuits). The various protective mechanisms provided at the level of the machine (hardware and microcode) are then presented : data access control and code execution control. Finally, a short presentation of the software production environment is given.

In conclusion, the compromises made between operational dependability and performance characteristics are summarized.

INTRODUCTION

The production of on-board computers is fraught with numerous and contradictory requirements. Operational dependability certainly constitutes one of the higher-priority requirements since missions assigned to the software are covering ever increasingly critical sectors (weapons management, countermeasures, flight controls, etc.). The presented CMF AIR computer constitutes the airborne version of the CMF range of computers developed under the auspices of the French DGA⁽⁴⁾. In this respect, it is certainly the computer of the CMF range required to satisfy the most severe requirements. In particular, it must meet the requirements defined for the mission computer of the ACE/RAFALE D aircraft which include :

- operational dependability (but a lesser degree than a flight-control management computer, for example),
- available space, weight and power consumption,
- high computing power,
- ease of extension,
- comprehensive instruction set, due to the variety of functions that have to be performed.

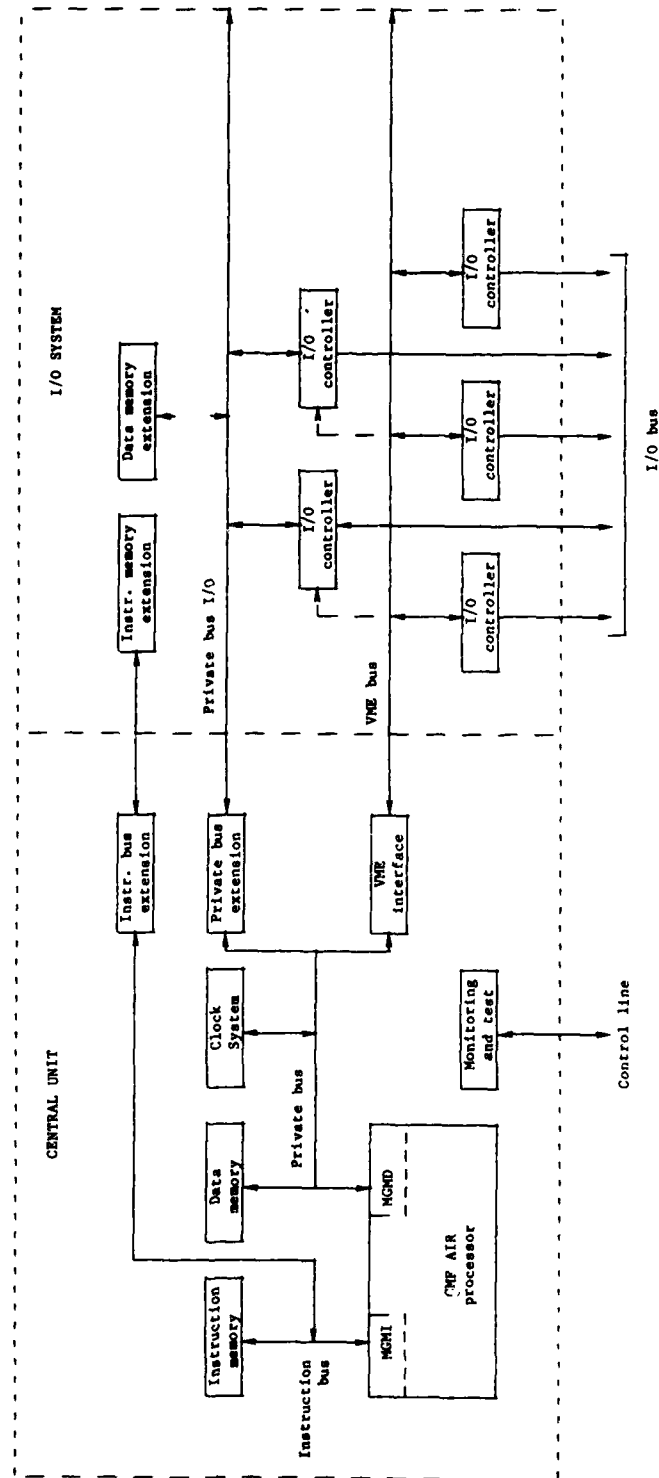
The design options of the CMF AIR computer therefore necessarily result from a compromise attempting to meet all these requirements as explained below.

CHAPTER 1 : GENERAL STRUCTURE OF THE COMPUTER

The CMF AIR computer consists essentially of a central unit, an input-output system and a power supply. This structure is illustrated in Figure 1 in a monocomputer configuration. It should be noted that in order to meet the requirements, a bicomputer solution is proposed, simultaneously providing the computing power and the high degree of reliability which are required by the mission.

- ¹ CMF : Calculateur Militaire Français - French Military Computer
- ² ACE : Avion de Combat Européen - European Combat Aircraft
- ³ Ada : Registered trademark of Dod/AJPO (Ada Joint Program Office)
- ⁴ DGA : Délégation Générale à l'Armement - French Ministry of Defense

Figure 1 : MONOCOMPUTER CONFIGURATION



The central unit is organized around two communication paths :

- the private bus, to which are connected :
 - . the CMF AIR processor
 - . the data memory
 - . the clock system
 - . the interfaces to the private bus extension and the VME bus
- the instruction bus which links the CMF AIR processor to the instructions memory.

1.1 CMF AIR PROCESSOR

A micro-programmed solution has been chosen, limited, for efficiency concerns, to one level of micro-programming. The micro-instruction format is 160 bits.

A four level pipeline structure is implemented, each level corresponding to one of the four main functions of the CMF AIR processor :

- instruction sequencing,
- fix/float operators,
- address generation,
- memory management (for both the instruction and the private bus).

The rationale for this organization is to provide :

- a simple (then efficient) architecture,
- a high level of functional parallelism between the address generator and the fix/float operator,
- protection against a pipeline deactivation when a control statement is executed,
- discrimination between instruction and data flows,
- deterministic execution times.

1.2 OTHER CENTRAL UNIT COMPONENTS

The private bus management is of master/slave type under the control of the CMF AIR processor address generator.

The clock system is in charge of the time reference at the system level. It also supports execution of "delay" statements at the application level. The clock system is equipped with a "watch-dog" device.

The interface to the (militarized) VME bus controls exchange of data on this bus. It also provides access to the private bus and to the data memory when requested by the I/O controllers.

1.3 TECHNOLOGICAL CHOICES

The central unit has been structured in functional blocks which are implemented in several Application Specific Integrated Circuits (ASICs). These functional blocks are listed below :

- instruction sequencing,
- program counter management,
- instruction memory management,
- address generation,
- execution processor,
- memory management,
- clock system,
- real-time management,
- timers,
- watch-dog,
- interface to VME bus.

The use of ASICs and the choice of C-MOS technology lead to implement the whole central unit in a single macro-hybrid on a 1/2 ATR board. The overall consumption is 20 watts. A significant benefit which derives from these technological choices is to ensure a high reliability at the component level (high degree of integration and low consumption).

1.4 EXTENSIBILITY

An essential feature of the selected design is to provide a high level of extensibility :

- at the memory level through the private bus extension,
- at the processing level by the possibility of using several central units in a multi-processor architecture.

CHAPTER 2 : RELIABILITY FEATURES

It is mainly at the level of the logic machine that specific mechanisms have been designed and implemented for improving operational dependability. The general structure of this logic machine is described, followed by a detailed description of the protective mechanisms.

2.1 MEMORY ORGANIZATION

The memory is organized into the following three types of zones :

- data zones,
- code zones,
- system zones (e.g. environment of a task).

System zones are entirely administered by the machine and invisible to the user.

Data zones are of the following five types :

- local zones containing the local data of the blocks, procedures and processes and parameters passed per value ; these zones are always allocated dynamically,
- visible and internal zones containing the visible (or internal) data of the modules ; these zones are always allocated before the start of application software execution ;
- heap zones containing the data created dynamically and addressed by pointers (instruction NEW),
- parameter zones containing the descriptors of the addressable parameters (dynamically allocated zones),
- static zones containing the remaining data structured into zones and allocated before the start of application software execution.

All these zones are addressed by means of a base and displacement, with the exception of the parameter zones which are addressed by means of a base and parameter number.

All the bases are automatically controlled by the machine (hardware and system software).

A base contains the following information :

- zone start address,
- length of zone,
- zone type and access rights.

Highly efficient execution checking is achieved by comparing this information with the operand access paths.

The code zones contain the whole of the program code (procedures and processes) executed in the modules (one zone per module). Each code zone is addressed by a base which specifies the start address and length of the zone. This base is used for initializing the program counter at each entry into the module or at each task activation. This structure allows highly effective checking of program counter evolution.

2.2 PROTECTIVE MECHANISMS

These mechanisms are provided by the logic machine (hardware and microcode). They apply to data access and code execution.

2.2.1 Data access checks

- 1) Address checks : these dynamically verify that an object has been correctly allocated within the zone assigned to it. These checks make use of the zone length and overflow information and can raise a ZONE_OVERFLOW basic exception.
- 2) Access rights checks : the data are assembled within zones by access types, thereby making it possible to check dynamically these access types.

Access rights are the following :

- for local zones : read and write, read and write limited to generation of the declarative part or transmission of the parameters and then read only, write only,
- for visible zones : read and write, read and write limited to generation and then read only, read and write from inside the module and read only from outside the module.

These checks may raise an `ILLEGAL_ZONE_ACCESS` basic exception.

- 3) Constraints checks : these checks, contrary to those described above, are performed by the software (code generated by the compiler).

They are performed when accessing the operands and use comparison operators. An example of such checks is the verification of value limits specified in language LTR3 or Ada. These controls are likely to raise a `NON_VERIFIED_CONSTRAINT` software exception.

It should be noted that the basic exceptions are processed by the logic machine while software exceptions are processed by the application software.

2.2.2 Code execution checks

- 1) Address checks : branch or internal procedure call instructions cannot modify the current base (external procedure calls and system calls only have this possibility). It is thus easy to check if the program counter overflows the code zone of the current module. In this case, a `CODE_ZONE_OVERFLOW` exception is raised.
- 2) Branch checks : dynamic monitoring of sequence breaks is performed by the machine by means of the following mechanism :
 - any procedure or service call must result in a procedure start instruction. If this is not the case, the `ILLEGAL_PROCEDURE_CALL` basic exception is raised. This exception is also raised if a procedure start instruction is encountered in sequence or following a branch,
 - any branch must be to a LABEL instruction. If this is not the case, an `ILLEGAL_BRANCH` basic exception is raised (a LABEL instruction encountered in sequence has no effect),
 - any exception referred to the software must activate an exception handler (standard or defined by the application software), necessarily starting with instruction `OPENH`. If this is not the case the `ILLEGAL_HANDLER` basic exception is raised. This exception is also raised if the handler start instruction `OPENH` is encountered in sequence or following a branch.
- 3) Procedure checks : the presence of a procedure start instruction is obligatory as the first instruction to be executed following the execution of a procedure call instruction. If this is not the case, the `ILLEGAL_PROCEDURE_CALL` basic exception is raised.

In addition, differentiation between procedure start instructions at the start of an internal procedure and at the start of an external procedure makes it possible to check that an internal procedure has not been called from outside. This check can raise the `ILLEGAL_PROCEDURE_CALL` basic exception.

CHAPTER 3 : SOFTWARE ENGINEERING ENVIRONMENT

Software executed on the CMF AIR computer will be primarily written in Ada, which by itself ensures a high level of reliability. Indeed Ada has been designed specifically for embedded software and features such as types, packages, exceptions, range of values and others are specially suited to the writing of highly reliable software.

But moreover, correctness of the software can be obtained through the use of tools, preferably integrated in a software engineering environment.

Such an environment already exists for the CMF AIR software : the ENTREPRISE environment, developed by the DEI⁽⁵⁾. ENTREPRISE is based on UNIX and includes an 'object management system' which is the repository for source and object codes. ENTREPRISE also includes LTR3 coding tools and a syntactic editor. Access control is supported ; access rights attached to users are checked each time an access to an object is made.

⁵ DEI : Direction Electronique et Informatique is a division of the French DGA (Ministry of defence)

But software reliability has to be 'constructed' all along the software life cycle and not only at the coding stage : a full set of integrated tools is needed. That is why the DEI is currently developing an enhanced version of ENTREPRISE which is multilingual (Ada, LTR3, ...), and includes the following tools :

- DLAO⁽⁶⁾ for software requirements specification,
- CLOVIS for software design,
- IDAS ⁽⁷⁾ for software testing,
- configuration management tool,
- project management tool.

Additional tools such as a documentation tool will be integrated later on.

A particular implementation of ENTREPRISE will be dedicated to aerospace applications : the SDA ⁽⁸⁾ software engineering environment which is developed by the ITI ⁽⁹⁾ Consortium under the auspices of the DGA.

We believe that, to improve software quality and especially software reliability, the effort has to be put on the software requirements specification and the software testing phases. This is consistent with the fact that these phases are the more time and money consuming. Therefore, we will focus on the related tools : DLAO and IDAS.

IDAS is already presented in the course of this conference.

DLAO is based on a semi-formal representation. A requirements specification is expressed in terms of objects (data, processes, events, states, ...) and relationships between objects. These objects are stored in a data-base. Taking advantage of this representation, DLAO provide efficient controls of the consistency and the completeness of the specification ; it also keeps an always up-dated version of this specification. Finally documents written in a subset of the french language are produced : easily readable, they are used as a contractual basis between the client and the software developer.

It has to be emphasized that quality at the specification stage is essential for the quality of the end-product that is its suitability to the users' needs.

CONCLUSION

The above description of the CMF AIR protective mechanisms shows that special attention has been paid in the design to operational dependability.

The advantages of such a choice are two fold :

- firstly, the operational dependability of the equipment is obviously improved. More specifically all the test mechanisms described above ensure high independence of software functions. Thus in the event of an anomaly concerning a non-critical function, this anomaly has no effect on other functions, thereby enabling the mission to continue,
- secondly, a structure of this type reduces the cost of software test, since the independence of functions considerably reduces the volume of regression testing which has to be performed following any modification.

It should nevertheless be emphasized that these advantages are achieved at the cost of a certain degree of performance degradation. Only operational use of the CMF AIR computer will be able to justify the compromises made between operational reliability and performance considering the aimed range of applications.

⁶ DLAO : Définition de Logiciel Assistée par Ordinateur (Computer Aided Software Requirements Specification) is a trademark of Electronique Serge Dassault

⁷ IDAS : Informatisation de la Détection d'Anomalies dans les Systèmes (Systems Anomalies Detection Computerization) is a trademark of Electronique Serge Dassault

⁸ SDA : Système de Développement Avionique - Avionics Development System

⁹ ITI : Integration du Traitement de l'Information - Data Processing Integration

TESTING EMBEDDED SOFTWARE

by

Michel Muenier

Electronique Serge DASSAULT

55, quai Marcel Dassault - 92214 Saint-Cloud - FRANCE

SUMMARY

This paper describes the techniques used by Electronique Serge Dassault (ESD) for testing embedded software, including in particular the mission computer software operated in the MIRAGE F1 and MIRAGE 2000 aircrafts.

An introduction presents the main characteristics of embedded software which make testing this software an activity of paramount importance.

A few figures present the ESD background in this field. Then the presentation of the MINERVE ⁽¹⁾ methodology used at ESD for developing software sets the scene for a more detailed description of each testing activity. The IDAS ⁽²⁾ and SVR ⁽³⁾ tools which support these activities are dealt with in the continuation of the paper.

Finally, some remarks are made on the evolution of testing techniques and a conclusion assesses the current situation and trends in software testing at ESD.

INTRODUCTION

Avionic applications and more generally embedded applications are characterized by a certain number of properties, such as long life-time and severe volume and weight constraints.

A certain number of factors specific to these applications have a particularly important influence on test activities. First of all, it is the high level of operational dependability required which, in itself, justifies the major part taken by test (40 % to 50 %) in the development process. Next, it is the "on-board" aspect which, because of the specificity of the target computers and their input-output systems, leads to host/target development and imposes operational environment simulation for tests performed on the target computers. In the same order of ideas, it is the "real-time" aspect which means that the software has to react in limited time to external events. At the test level, this "real-time" characteristic requires the software environment to be reproduced as faithfully as possible and the execution of the tested software not to be disturbed (e.g. by instrumentation).

Other less specific factors, such as the high modification rate and the concern for productivity, also have considerable influence on the testing of avionic software (leading respectively to regression testing and symbolic testing).

All these factors make the testing of embedded software a difficult and costly operation for which few tools were available only a few years ago. It was then said that real-time applications constituted the "lost world" of software test and debugging (Robert L. GLASS, Boeing Aerospace Company).

This paper describes the solutions which, at Electronique Serge Dassault (ESD), are being and have been applied to these problems for several years.

CHAPTER 1 : EXPERIENCE OF ESD

ESD has specialized in the design, development and manufacture of electronic equipment for both civilian and military applications.

ESD supplies numerous equipments for the MIRAGE F1 and MIRAGE 2000 aircrafts produced by Avions Marcel Dassault-Breguet Aviation (AMD-BA) for which it has developed several ranges of general-purpose airborne computers (the M182 computers for the MIRAGE F1 aircraft and the 84 series computers for the MIRAGE 2000 aircrafts). Since 1977, ESD has also been developing the operational software executed on these computers as well as the associated production and test software. In order to meet these requirements, the Company has established large software engineering teams (more than 200 people).

¹ MINERVE : Méthodologie Industrielle pour l'Etude, la Réalisation et la Validation de logiciel d'Équipement (Industrial Methodology for the Study, Production and Validation of Equipment Software), is a trademark of Electronique Serge Dassault

² IDAS : Informatisation de la Détection d'Anomalies dans les Systèmes (System Anomaly Detection Computerization) is a trademark of Electronique Serge Dassault

³ SVR : Software Validation Rack

More than 20 million bytes of operational software have thus been delivered for a total of 15 projects. A delivery is made for each project approximately once a month, comprising on an average 600,000 bytes of code, 60,000 pages of listings and 10,000 pages of documents.

90 % of this software is written in LTR, a real-time Pascal-like language. The remaining 10 % of this software corresponds to sections for which program execution time is critical and which are therefore written in assembly language. Since 1987, ESD is also equipped with tools (compilers and testing tools) for applications written in Ada [MUE 87].

Emphasis should be placed on the major effort undertaken by ESD over the last ten years in the area of software engineering, which has resulted in the definition and implementation of MINERVE methodology and in the development of tools : DLAO⁴, IDAS and SVR. The next chapter briefly describes the MINERVE methodology.

CHAPTER 2 : MINERVE METHODOLOGY

The objective of MINERVE is to facilitate the production of high-quality software under controllable conditions of cost and lead time.

The quality of software is not only determined by its conformity to requirements but also by its ease of modification, clarity, accuracy of its documentation, operational dependability, performance characteristics, etc.

Cost and lead-time control depends on the possibility of undertaking at all times action contributing to the execution of contractual obligations. Such control is achieved by permanent knowledge of project progress and work outstanding.

In order to achieve the above objectives, MINERVE, a key element for software quality assurance and control within ESD, is based on the following three principles :

- projects are divided into phases and stages marked by clearly defined activities, products and responsibilities,
- the quality of products as well as their costs and lead times are monitored in a continuous manner,
- modifications may be included whatever of the degree of progress in accordance with a single procedure intended to avoid any degradation of software quality.

2.1 PRINCIPLE OF PROJECT BREAKDOWN

Projects are broken down chronologically into two levels (phases and stages) in accordance with the following scheme :

PHASE 1 - DEFINITION

- Stage 1.1 : Overall Definition of the System
- Stage 1.2 : Operational Definition of the Software
- Stage 1.3 : Functional Definition of the Software

PHASE 2 - PLANNING

- Stage 2.1 : Checking of Technical Feasibility
- Stage 2.2 : Definition of Technical Means
- Stage 2.3 : Reexamination of the Software Quality Plan
- Stage 2.4 : Reexamination of Planning and Costs

PHASE 3 - PRODUCTION

- Stage 3.1 : Basic Design
- Stage 3.2 : Detailed Design
- Stage 3.3 : Coding and Unit Tests
- Stage 3.4 : Integration Tests
- Stage 3.5 : Functional Tests
- Stage 3.6 : Software Validation

PHASE 4 - OPERATION

- Stage 4.1 : System Integration
- Stage 4.2 : Software Maintenance

Stages 1.1 and 1.2 and Phase 4 are the responsibility of the weapon system contractor, while all other stages are the responsibility of the software producer.

⁴ DLAO : Définition de Logiciel Assistée par Ordinateur (Computer-Aided Software Specification), is a Trademark of Electronique Serge Dassault

A STAGE is characterized by the nature of its MAIN ACTIVITY. The latter always results in one or more formal products or MINERVE PRODUCTS : documents, files, magnetic tapes, etc. Once a main activity has been completed, its products can be modified only by a special procedure (see par. 2.3).

2.2 PRINCIPLE OF CHECKING

Checks are made throughout the project at the following two levels :

- quality of products (programs and documents),
- cost and lead time.

2.2.1 Quality Control

Any quality control is performed at the level of a MINERVE product.

These controls are of the following three types :

- Type A Control

This type of control consists of performing internal checks on the product of a stage. These checks verify that this product obeys the specific rules (defined in the software quality plan) and respects the general standards (defined in the software quality manual) applicable to it.

- Type B Control

This type of control consists in checking the coherence between the products of a stage and those of former stages. This type of control, as for the previous type, takes the form of rereading documents and project reviews.

- Type C Control

This type of control consists of program testing which is performed in four successive stages. The tests performed during the stages of coding and unit tests (3.3), integration tests (3.4), functional tests (3.5) and software validation tests (3.6) check the conformity of the programs with their successive descriptions established in the course of the 'symmetrical' stages in the life cycle (cf. figure 1) : detailed design (3.2), basic design (3.1), functional definition (1.3) and operational definition (1.2).

These types of controls are illustrated by Figure 1.

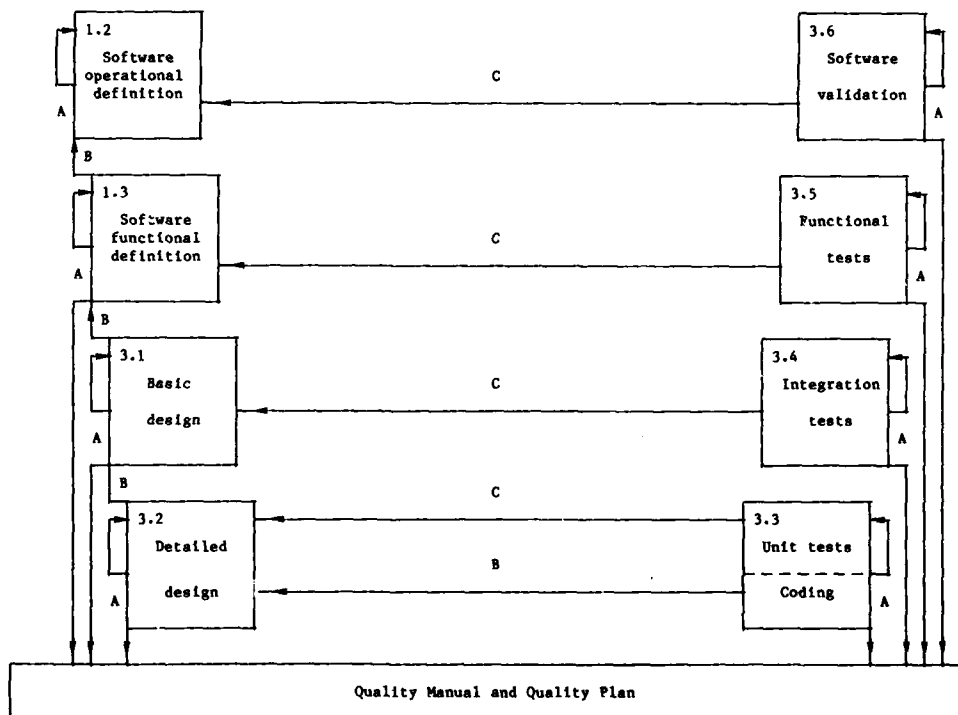


Figure 1. SOFTWARE LIFE-CYCLE AND QUALITY CONTROLS

2.2.2 Cost and Lead-Time Control

This control is applied at close intervals throughout the period of the project. It consists in evaluating the degree of work progress in terms of cost and time and then comparing it against forecast information recorded in a reference document.

2.3 PRINCIPLE OF MODIFICATION

Any modification can be accepted irrespective of the degree of project progress.

No MINERVE product can be modified outside of the procedure described below and which is essential for maintaining coherence between the various products throughout the life cycle of the software.

- Acquisition of a Modification

After determining the MINERVE product to be modified located the furthest upstream in the software development process, a MODIFICATION SHEET for this product is completed by the customer if it relates to products of stage 1.2 or by the software supplier if it relates to products of a later stage.

- Execution of a Modification

The modification applied in this manner results in rework of all the stages whose products are involved, starting with that furthest upstream.

The application of modifications induced in the various products concerned is recorded on a SUPERVISION SHEET raised as soon as the decision to execute has been taken.

CHAPTER 3 : SOFTWARE TEST

As described in the previous Chapter, software testing occurs during stages 3.3 to 3.6 of the life cycle. The present Chapter relates only to actual software test, excluding program inspections, walk-throughs and reviews [MYE 78] which are performed during design and coding stages.

For stages 1.2 to 3.2, the system has been divided into different levels (operational functions, software functions, modules, parts, etc).

At each stage, the tests therefore apply to the breakdown level described in the "symmetrical" stage (see Figure 1) in accordance with a process of progressive recomposition of software "blocks" of increasing size. This approach has the advantage of facilitating environmental simulation at all stages.

When larger assemblies of software are tested, the combinatory explosion leads to abandon testing information internal to programs, giving way to functional tests which become more significant. The corresponding test techniques are commonly identified by the terms "white box" test and "black box" test.

Use of white box and black box tests is summarized in figure 2 :

| STAGE | | WHITE BOX TEST | BLACK BOX TEST |
|-------|---------------------------|---|-------------------------------------|
| 3.3 | Unit tests | Branch tests Path tests | External part tests |
| 3.4 | Static integration tests | Branch tests Inter-part interface tests | External module tests |
| | Dynamic integration tests | Inter-process interface and synchronization tests | |
| 3.5 | Functional tests | | External software function tests |
| 3.6 | Validation | | External operational function tests |

Figure 2. TESTING METHODS

It should be emphasized that test activity (identifying a malfunction) is associated with a debugging activity (localizing the error(s) which cause(s) the malfunction). Debugging necessitates the exploitation of information internal to the program (approach of the "white box" type).

A more detailed description of the test techniques used in stages 3.3 to 3.6 is given in the following paragraphs.

3.1 UNIT TESTS

The breakdown level concerned is the "part", whose characteristics are its small size (less than 100 instructions), to correspond to a programming language structure possessing a single entry point and a single exit point (e.g. a procedure) and to be a compilation and archiving unit.

Unit tests generally include a pattern of entry data and a pattern of expected output data. In this regard they may be viewed as using a 'black-box' approach (external part tests). But they also consider internal characteristics of the 'part' such as 'branches' and 'nodes' (see below) and therefore belong equally to the 'white-box' class of tests.

Unit tests have to ensure a maximum reliability

- the entry data pattern must lead to the execution of each 'branch' of the part. A 'branch' is a sequence of statements between two 'nodes' (if, case, ... statements). The entry data pattern must also validate to a large extent that there is no correlation between the nodes (i.e. that a node in a given state (exit branch) does not 'freeze' the state of another node). This lead to test a set of paths which number is expressed by the 'cyclomatic number' [MCA 76] It should be noted that an execution of all possible paths is generally too costly while not improving significantly the level of coverage,
- unit tests executions must be independent : entry data patterns must not be generated by previous executions of the same test.
- software robustness must be checked. This means that entry data patterns have to include values covering the normal range of parameters but also values outside this range. It must be also controlled that no 'side-effect' is induced : the tester has to verify that the 'environment' of the part (such as entry data) is not altered by the execution of the test,
- unit test must check that execution time and memory space constraints (as expressed in the SOFTWARE DETAILED DESIGN DOCUMENT) are satisfied.

Unit tests have to be maintainable

- functional test description, test operations and test results have to be archived in UNIT TEST FILES, MINERVE product of the unit tests stage. A modification of a part will generally cause an update of the corresponding UNIT TEST FILE, so that the test case remain consistent with the code.
- UNIT TEST FILES will be used to perform regression testing i.e. reexecuting a set of unit tests whenever a part is modified (see IDAS tool, chapter 4.1).

3.2 INTEGRATION TESTS

The breakdown level concerned may be a process or module.

A process corresponds to the set of programs whose execution is subject to the same activation condition (external or internal event, frequency).

A module is the reusable component of software between different versions of an application (e.g. for a combat aircraft, versions related to different missions or armaments).

It belongs to a unique process, has a functional consistency and is composed of an interface part and a body part. The interface expresses the visibility from and towards the module, while the body part comprises local data and statements (part calls). The latest programming languages (LTR3, Ada) now represent process and module structures very simply.

Integration tests are of two kinds :

- STATIC INTEGRATION TESTS : they consider a set of paths inside a process or a module and check, with respect to the SOFTWARE BASIC DESIGN document (MINERVE product of the basic design stage) :
 - . the interface between parts and modules (i.e. consistency between formal and actual parameters in parts calls),
 - . the functional aspect of modules (a black-box type of test).
- DYNAMIC INTEGRATION TESTS : they check the interfaces between processes and the external environment, and between the processes themselves. They also control time sharing (cycle duration, synchronization, etc.) and data management (access to shared variables, etc.).

As for unit tests, integration tests are archived in INTEGRATION TEST FILES, MINERVE product of the integration test stage.

3.3 FUNCTIONAL TESTS

The purpose of this stage is to check the conformity of a 'software function', against the SOFTWARE FUNCTIONAL SPECIFICATION document (a MINERVE product of the software functional definition stage).

A 'software function' is the first level of breakdown of the overall software. It may be related to an operational point of view (e.g. Air-to-ground function) or to an internal point of view (e.g. Digital bus management function).

Functional tests are archived in FUNCTIONAL TEST FILES, MINERVE product of the integration test stage. They include :

- a sequence of actions described in functional terms and as a sequence of instructions specifying how to operate the tools,
- a set of checks also described in a dual point of view. It should be noted that some of the checks are performed manually, using 'images' displayed on the screen by the SVR tool (see chapter 4.2).

The following three types of test are executed :

- nominal tests, corresponding to cases of normal weapon system operation,
- fault tests, corresponding to cases defined at the definition stage,
- random fault tests with the aim of evaluating the robustness not only of the program but also of the functional specifications, thus possibly questioning the latter.

3.4 SOFTWARE VALIDATION

The tests performed for validating the software are similar to the functional tests but apply to the overall program and are executed with respect to the SOFTWARE OPERATIONAL SPECIFICATION and SOFTWARE INTERFACE SPECIFICATION documents (MINERVE products of the software operational and software functional definition stages). At this stage, regression tests are applied systematically.

Validation tests are archived in the VALIDATION TEST FILES, MINERVE product of the stage. They constitute a contractual basis for software acceptance between the prime contractor and the software developer.

CHAPTER 4 : TOOLS

The tools used at various software test stages are figured below :

| | | |
|-----|---------------------|----------|
| 3.3 | Unit tests | IDAS |
| 3.4 | Integration tests | IDAS |
| 3.5 | Functional tests | IDAS SVR |
| 3.6 | Software validation | IDAS SVR |

For unit tests and integration tests, a static version of IDAS is used. It is a test harness that provides the ability to express the data entry pattern and the test actions in terms of a test program using a dedicated language. But at this stage real-time aspects are either ignored (unit tests and static integration tests) or only simulated (dynamic integration tests). The test programs and the tested programs are run both on the host computer.

For functional tests and validation tests, a dynamic version of IDAS is used (cf. fig. 3). This version runs on a dedicated machine separated from the target machine. It provides the essential capability of observing the dynamic behaviour of the tested program without any disturbance. But it does not provide the dynamic simulation of the environment. This is performed by the Software Validation Rack (SVR).

4.1 IDAS

IDAS [MUE 85] is a software test system specially designed for real-time applications. The mode of operation consists in executing the tested program, observing its behaviour and checking it with respect to a reference behaviour [AYA 79]. IDAS can adapt to a large range of programming languages (LTR, Ada, etc.) as well as various target computers on which the tested program is executed.

IDAS is based essentially on the use of a test language allowing to write test programs which may be stored and reexecuted (regression testing).

Test programs allow handling at the symbolic level of the variables and instructions of the tested program. For this purpose, IDAS is connected to the coding tools (compilers, assemblers, etc.) via a post-processor which generates a data base of tested objects descriptors.

The static version of the IDAS system uses a target computer simulator instead of the actual embedded computer.

- The general structure of the dynamic version of IDAS is illustrated by Figure 3.

- an interpreter used for debugging the test programs,
- a compiler used for executing the test programs in real-time,
- a debugging tool used for fixing errors in the tested program.

- a modelling tool which enables to express a behaviour model, using the Petri nets formalism [LAM 85],
- a simulation tool which enables to substitute 'templates' to instructions or procedures.

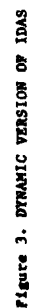


Figure 3. DYNAMIC VERSION OF IDAS

4.2 SOFTWARE VALIDATION RACK (SVR)

The SVR is an assembly of hardware and software used for dynamic testing of real-time software.

The functions performed by the SVR consist in simulating the environment of the target computer, and checking the behaviour of the embedded software through the operational links.

The simulation takes into account an aircraft flight path, parameters of which are calculated and recorded in a computing centre. The use of a graphics console allows simulation of control unit front panels and operational displays. In particular, it allows the simulation of operational actions (e.g. actions on the control panels) and environment modification (e.g. failure of an equipment). This console also displays the information carried over the operational links.

The tests are executed either manually or automatically. In the former case, the simulation of operational actions and environment modification is performed by the operator. In the latter case, the simulation is derived from a test executed previously, known as a "reference test". During this test the list of parameters to be checked and the reference values are recorded. This information may be manually modified in the course of test execution to take into account the impact of the software modification on the scenarios.

The hardware structure of the SVR is illustrated by Figure 4.

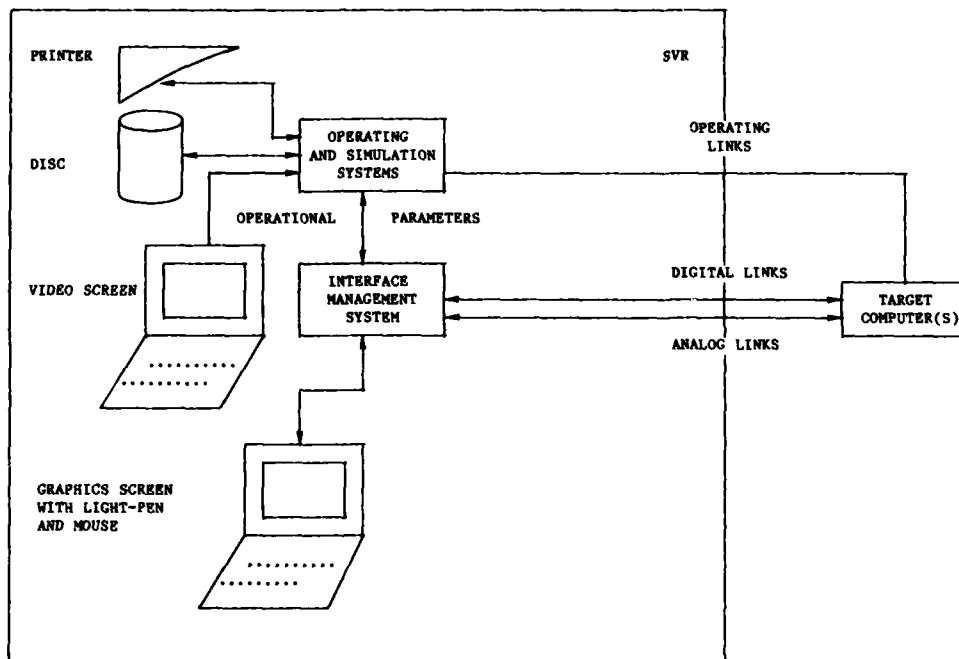


Figure 4. SOFTWARE VALIDATION RACK ARCHITECTURE

The SVR hardware consists of two main assemblies :

- an operating and simulation system connected to :
 - . the operating link for the target computer (computer stop/restart),
 - . standard peripherals : printer used for editing anomalies, disc on which data required by the automatic tests (flight parameters, reference parameters, test results) are stored and a video screen/keyboard for system operation.
- an interface management system connected to :
 - . the target computer(s) via operational links
 - . a graphics screen with mouse and light-pen, used to display images of operational interfaces (control panels, head-up, head-down, ...)

The SVR software consists of the following four main assemblies :

- flight parameter generation software :
It generates flight parameters transmitted to the SVR in a form compatible with the operational digital links.
This software is run on equipment distinct from the SVR (IBM computing center).
- SVR operating software :
It presents a dialoguing interface in the form of programmable function keys implementing arborescent menus. This software includes :
 - . static debugging of the target computer software,
 - . validation/acceptance test with environment simulation, including selection of various operating modes (normal, slow motion, step by step at the cycle level) and the selection of displays and presentations (zoom, ...).
 This software is executed on the operating and simulation system.
- environment simulation software :
It ensures the simulation of each equipment transmitting data to the target computer(s). This simulation is limited to the functions generating these data.
This software is executed by the operating and simulation system.
- automatic test software :
It has the following two purposes :
 - . to allow recreation of test configurations, which is useful for analyzing malfunction during functional test and validation,
 - . to automate test operation and results analysis. This function is extensively used for acceptance procedures.
 This software is executed on the operating and simulation system.

CHAPTER 5 : TRENDS IN SOFTWARE TESTING

Short and medium-term evolution of software tests do not relate as much to actual test methods and tools as to integration of the test process with other activities of software development.

The present Chapter is therefore mainly concerned with the effect due to the appearance of tools used earlier in the software life cycle (mainly software specification aid tools) and to the present tendency of integrating tools within Software Engineering environments.

5.1 TEST AND SPECIFICATION METHODS

Awareness of the exponentially increasing cost of modifications during the course of development has prompted over the last few years considerable work in the area of software definition (and to a lesser degree software design).

At ESD, this effort has resulted in the production of the DLAO tool allowing software requirements specifications to be expressed in a semiformal manner. The basic concepts made available to the user are :

- information representing the operational data (e.g. mode of radar operation),
- interfaces, the physical support of data (e.g. encoding used for transmitting over a multiplexed bus),
- events which modify the set of activated processes (e.g. change of mode of radar operation),
- states which characterize software operation at a given instant (e.g. air-to-ground mode),
- processes corresponding to elementary tasks (e.g. ballistic calculations).

The data defined in this manner by the user as well as the associated comments are stored in a data base which thus always contains an updated representation of the software requirements specifications. Many possibilities of producing a document are offered, exploiting the possibilities of data base queries.

Such specification formalization should allow in the fairly near future significant progress in test automation. ESD is conducting work in this direction, aimed at the automation of test scenarios generation. The approach adopted consists in generating a prototype from a subassembly of the requirements specification (defined by the user) and then, by means of input stimuli, in producing output data by exercising this prototype.

The input stimuli, which constitute the scenario input data, can be generated manually or semi-automatically from the requirements specification. In the latter case, the user has the possibility of limiting the information involved in the scenarios in order to avoid the phenomenon of combinatory explosion. The output data generated by the prototype constitute the reference value used when executing the test scenarios.

5.2 TOOL INTEGRATION

The new awareness of Software Engineering concepts is leading to increasingly higher integration of tools. A significant example is the work presently undertaken by the ITI ⁽⁵⁾ Consortium of eight French companies of the aerospace sector. This work aims at producing software environments dedicated to avionic systems and on-board software development.

Such environments are providing powerful mechanisms allowing the integration of tools, such as :

- a software requirements specification tool,
- a design tool,
- coding tools,
- testing tools.

These mechanisms consists essentially of :

- an "object" data base, standard means of communication between tools,
- a configuration management system ensuring permanent coherence between the objects of the base (requirements specifications, design documents, code, test programs).

Better integration between the test tools themselves (SVR and IDAS) will also be ensured thereby making it possible from the same work station to control simultaneously the operations of functional and validation test performed by the SVR and debugging operations (in particular, the elimination of faults related to real-time problems) performed by IDAS.

CHAPTER 6 : CONCLUSION

Our experience of avionic software testing allows us to present a largely positive situation.

The main objective, i.e. software quality, has been achieved : less than one error per year and per program has been found in flight.

We are also maintaining control of our costs and lead times in spite of the high rate of evolution of our software.

We have also noted continuous growth of our productivity, which has increased by a factor of nearly five between 1979 and 1987. This should be weighted by increased use of computer resources at the same time, but the overall balance is positive.

Our efforts today are tending to maintain this growth in productivity and we hope to achieve this objective in the following two main ways :

- continuous effort at the unit test and integration test level since, as emphasized in Chapter 3, the earlier errors are detected in the life cycle, the less they cost to correct. In particular, we intend to perfect the methods and tools allowing us better control over the coverage factors of these tests,
- larger-scale integration of our development tools within software engineering environments (including the use of multi-window graphics work stations) and their continuous adaptation to avionic system design methods.

REFERENCES

- [AYA 79] AYACHE - AZEMA - DIAZ
Observer : a concept for on-line detection of control errors in concurrent systems
9th International symposium on fault tolerant computing
MADISON June 1979
- [BAR 82] P. BARBIER - S. CHENUT - MARTIN - F. DOLADILLE
DLAO : Définition de Logiciel Assistée par Ordinateur
Premier colloque de Génie Logiciel
APCET
Paris Juin 1982
- [LAM 85] G. LAMARCHE - P. TAILLIBERT
Utilisation des réseaux de Petri pour la test des programmes temps réel
Technique et Science Informatiques (TSI) Volume 4 n° 1
Janvier-Février 1985

⁵ ITI : Intégration du Traitement de l'Information (Data Processing Integration)

- [MCA 76] T Mc CABE
A complexity measure
IEEE Trans. on Software Engineering Vol SE-2 n° 4
pp. 308-320 - December 1976
- [MUE 85] M. MUENIER
IDAS : an automated software test system for real-time applications
Overview of the system
NSIA Joint Conference on Software Verification and Testing
Colorado Springs October 1985
- [MUE 87] M. MUENIER - P. DEMCKER
The Ada-IDAS project
Ada-Europe International Conference
Stockholm May 1987
- [MYE 78] G. MYERS
The art of software testing
Wiley - Interscience
Business data processing serie 1978

ADVANCED AVIONICS ARCHITECTURE
PAVE PILLAR

JOHN C. OSTGAARD

U.S. AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AVIONICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OHIO, U.S.A. 45433

SUMMARY

The increased use of advanced electronics has given modern combat aircraft phenomenal levels of performance, but at a stiff price in initial cost, maintenance workload and aircraft availability. Hence, aircraft design is shifting to give equal emphasis on performance, affordability, maintainability, and reliability in the development of avionic systems. This paper discusses the challenges and benefits of an avionics architecture concept which integrates avionic functions at the system level thereby improving the system's accuracy, increasing its immunity to failure, and decreasing its reliance on multiple redundant sensors. This design philosophy, which permits resources to be shared across subsystems, requires a highly coupled system-wide management and control program (operating system) supported by a wide-band data distribution network, high-speed data and signal processors, and extensive mass memory. The implementation strategy for this avionics architecture is the system-wide utilization of common modular building blocks using advanced microelectronics such as VHSIC, standard 3/4 ATR modules and integrated racks, and interconnected by fiber optic networks.

The generic integration approach and architecture produced by the PAVE PILLAR program is the foundation for avionics development in next generation aircraft for the U.S. Department of Defense and include such aircraft as the USAF Advanced Tactical Fighter (ATF).

INTRODUCTION

In performing its role of research and development innovator for aviation electronics, the Avionics Laboratory, an active member of the Air Force Wright Aeronautical Laboratories, has initiated a major program to develop the concepts for the next-generation integrated avionics system architecture. The program is called PAVE PILLAR and its purpose is to achieve significant improvements in the availability, cost of ownership and mission effectiveness of future combat aircraft.

Current avionic systems are organized by functional areas. Typical of these are flight control, stores management, electronic warfare, navigation, weapon delivery, communication, electrical power control, engine control, dedicated controls/displays, etc. The systems integration which exists is limited to the subsystem or intrafunctional level. This subsystem approach arose partially because of the lack of technology to support a total systems integration approach and partially because of traditional political boundaries.

These unique subsystems, some of which have been developed at the eleventh hour to satisfy an operational deficiency, put a tremendous strain on the logistics support structure of the Air Force. This is because each unique subsystem has numerous unique line replaceable units (LRUs), all of which require spares, which results in an extremely high life cycle cost for the weapon system.

EARLY INTEGRATION APPROACH

With the adoption of MIL-STD-1553B, Aircraft Internal Time Division Command/Response Multiplex Data Bus, some of the newer U.S. Air Force strategic and tactical aircraft have begun the task of integrating beyond the subsystem level. However, even these newer system integration activities have addressed only the basic avionics functions of navigation, weapon delivery, communication, and to some extent, controls and displays. Analysis of a current tactical aircraft shows that the avionics suite consists of approximately 58 line replaceable units with 437 different sub-assembly spare types. The 1553 data bus and dedicated cables alone require 254 harnesses to connect the external line replaceable units. The combined number of internal and external interconnects explodes to more than 86,600 connections. Forty percent of the maintenance actions on the flightline are concerned with these cables and connectors.

Analysis shows that by applying high speed fiber optic multiplexing and Very High Speed Integrated Circuits (VHSIC) technologies, the number of cables and connectors can be reduced by as much as thirty-five percent. However, this does not imply that one can just insert these technologies and receive the improvements. One must structure or partition the technologies into a system architecture which will permit maintenance personnel to easily isolate faults and to replace failed avionics components. Also, mission analysis has shown that the system architecture must be flexible enough to support air-to-air and air-to-ground mission requirements. This leads one to conclude that the system architecture must adapt during the life-time of the system to constant changes and new mission requirements.

It follows then that the hardware and software which form the system architecture must have a modular basis which will permit the maintenance personnel to remove and replace the components, yet allow system designers to adapt the avionics suite to new requirements. PAVE PILLAR was created to solve these problems.

EVOLUTIONARY PROCESS

The PAVE PILLAR program is an evolutionary process whereby an avionics architecture is developed using revolutionary new technology. The trend in avionics system design is to increase system integration thereby improving system accuracy, increasing system immunity to failures, and decreasing system reliance on multiple redundant sensors. This design philosophy which permits resources to be shared across subsystems requires a highly coupled system wide management and control program (operating system) supported by a wide band data distribution network, high speed processors, and extensive mass memory. The ability to consider integration of this magnitude is dependent on recent advances in microelectronics technology, most notably the advent of VHSIC technology.

The PAVE PILLAR avionics architecture is functionally divided into three distinct areas:

Mission management

Sensor management

Vehicle management

The three areas define the enclosing boundaries for resource sharing, sparing, and substitution. Unique characteristics of each of these areas preclude the utilization of resources across areas for the purpose of function recovery or reconfiguration. This does not imply that the areas do not contain many common hardware elements, but that the organization, connectivity, and control of these components restrict their practical use in functional system-wide reassignment. Each of the three areas have associated with them a logical processor type and varying interface requirements. Figure 1 illustrates the top level organization of the PAVE PILLAR avionics architecture, system elements are built from a set of common modules supporting programmable processing, I/O, and memory storage functions. The interfaces between system elements are standard high speed time division multiplex buses and data links, all utilizing fiber optics technology.

MISSION MANAGEMENT AREA

The Mission Management area consists of Mission Data Processors, Mission Avionics Multiplex Bus, Block Transfer Multiplex Bus, System Mass Memory, Stores Management System, and a collection of interfaces to the Mission Avionics Bus. This area provides the resources to perform the mission and system management functions such as fire control, target acquisition, navigation management, defense management, stores management, TF/TA/OA functions, and crew station management. The Mission Management Area contains identically configured Mission Data Processors all connected to the Mission Avionics Multiplex Bus and loadable from the System Mass Memory via the Block Transfer Multiplex Bus. All Mission Processing control and data exchange outside of loading operations is performed using the Mission Avionics Multiplex Bus. The Mission Management Area controls job assignments for the Signal Processors and determines the connection paths from Signal Processors to sensor front ends. High level processed information is collected from the Signal Processors (such as target tracks, CMI reception results, threat descriptions) and coding and control data is provided back to both the Signal Processors and the sensor front ends.

The Mission Management Area also interfaces with the Vehicle Management Area to receive navigation state information and to supply route and trajectory commands. Other interfaces are to cockpit multi-function switches, Stores Management System, and miscellaneous avionics control devices not directly interfaced to the Mission Avionics Bus (helmet mounted sight, voice recognition system, data recorders/readers). The Mission Management Area collects the health and status of all core elements and sensor/subsystem components for maintenance history and to maintain mission functional capability.

SENSOR MANAGEMENT AREA

The Sensor Management area consists of a set of common Signal Processors, a sensor data distribution network, a sensor control network, a data exchange network, and a video distribution system. The sensor management area provides the signal processing functions and interfaces necessary to convert conditioned data from multiple sensors via a sensor network into processed information suitable for distribution to other avionics systems. The sensor management area accepts encrypted data from a TRANSEC/COMSEC controller(s) and processes the data for transmission. This area also distributes processed digital video to crew displays and distributes sensor control commands to sensors via a sensor control network. Signal Processor task assignment, sensor data distribution network control, and Signal Processor resource reconfiguration is managed by the Mission Data Processors.

VEHICLE MANAGEMENT AREA

The Vehicle Management System (VMS) is an independent subsystem supporting the fundamental flight and airframe related control functions. The VMS Management area is physically isolated from the rest of the architecture for safety of flight reasons and contains a higher degree of physical redundancy, usable only within the VMS area. The VMS management area contains VMS Data Processors, Controls/Displays interfaces, Flight Sensor/Actuator Interfaces, Electrical Power Control Interfaces, Engine Control Interfaces, and Utility Systems Interfaces. The VMS Data Processor is essentially identical to the Mission Data Processor except for memory configuration, multiplex bus interfaces, and reconfiguration methods. The VMS Data Processor has Read Only Memory for all program storage and does not perform any program loading. The VMS Data Processor also has six High Speed Bus interfaces. Two dual redundant interfaces to the Mission Avionics Multiplex Bus and four simultaneously active bus interfaces to the VMS Multiplex Bus. Each VMS Data Processor contains the entire program load to perform all VMS functions. Reconfiguration is accomplished by activating

dormant tasks in available spare VMS Data Processor resources. The flight essential nature of VMS processing necessitates a very high degree of functional reliability (Fail-Op/Fail-Op/Fail-Safe). This is accomplished by physical quad redundancy of sensors, buses, processors, and actuators in the VMS processing area. The partitioning of processing functions among VMS Data Processors retains the quad redundant, simultaneously active characteristic of the VMS area.

SYSTEM CONTROL

Control of the core processing system is provided by a distributed software architecture providing for commonality of control software across the mission processors, flight control processors, and signal processors. Major control functions include:

- Initialization and system startup and restart
- Assignment of application software task to processing resources (software configuration and reconfiguration computing resources management)
- Sequencing and synchronization of related software tasks
- Management of sensor and other device resources with respect to mission objectives, mode and task management, and software parameters
- Interpretation of response to, and integration of, human control into the system functionality
- Collection, maintenance, and reporting of system hardware and software status, and operational functionality
- Response to hardware and software failure detection to preserve mission effectiveness
- Flight control change management and response
- Reintegration of recovered hardware or software functions
- Assurance of a distributed data base consistency and integrity, management of access to that shared data
- Preservation and collection of data required for continuity of system functionality across failure recovery points
- Management of communications access to ensure optimal use of communication resources and correct addressing of data messages
- Assurance of the security of classified data

The operating system is partitioned into three elements: (1) the kernel executive which provides those functions common to all processors, (2) the distributed executive which provides for decentralized system control in each processor, (3) system executive which provides the monitoring of system state and reconfiguration based on mission requirements and detected system failures. Figure 2 depicts the interrelationship of the three elements.

OPERATIONAL CONCEPT

The PAVE PILLAR architectural concept was developed to support aircraft operations from deployed locations with a minimum of support. This architecture supports the resource sharing of core data and signal processing resources and is constructed of a set of common modules that specifically support a two-level maintenance concept. This architecture supports high degrees of systems availability and reliability. This is accomplished through the application of spare signal and data processing resources at the system level so that backup services are provided when the primary sources fail. In addition, the architecture supports graceful degradation in that when spare resources are exhausted remaining resources can be assigned to the highest priority functions on a mission basis.

COMMON MODULES

The PAVE PILLAR architecture is physically comprised of a number of "building blocks" called common modules. A common module is sized to contain the circuitry to perform a complete digital processing function including interface control and health diagnosis. The approach for PAVE PILLAR is to develop common modules from a limited VHSIC chip set and then develop systems/subsystems which utilize the common modules. The exact composition of the VHSIC chip set will depend on the evolution of that technology program, while the members of the common module family will be subject to PAVE PILLAR avionics system design considerations.

A number of common modules can be built up from a family of VHSIC chips which in turn can be grouped to form the basis for any one of the avionics subsystems depicted in Figure 3. Certain non-common modules will undoubtedly be required for some specific subsystem implementation, however, the reduction in numbers of spares types required as a result of common module usage will provide a significant cost and effectiveness improvement.

The Avionics Laboratory has undertaken the design and development of two common module sets; the VHSIC 1750A Data Processor and the VHSIC Common Signal Processor. These modules are being designed with

stringent requirements for both extremely low failure rates and high fault detection and fault isolation capabilities. Fault isolation to the single module level will be accomplished by on-board Built-In-Test (BIT) circuitry at the chip level and multi-tiered self test software. The use of a modular concept will permit the maintenance personnel to perform on-board diagnosis and replacement of the avionics at the module level with no auxiliary ground equipment.

This capability then leads one to conclude that a two-level avionics maintenance concept might be realizable. Figure 4 shows the impact of various maintenance concepts. The current three level maintenance approach consists of flight line, Avionics Intermediate Shop, and depot/factory diagnosis and repair. To illustrate the benefits of changing from a three-level to a two-level maintenance concept an in-house study to examine the relative life cycle cost of several combinations of maintenance concepts and technology integration was conducted. As shown here, the costs consist of operations and support plus the Avionics Intermediate Shop and spare parts to support 1000 aircraft for 20 years at a flying rate of 300 flight hours per aircraft per year. All costs are stated in FY84 dollars. The first column represents today's technology - F-16 A/B - utilizing the standard three-level maintenance with removal/replacement of LRUs at the flight line. In column B the three-level maintenance concept is retained and LRUs are still used, but VHSIC technology has been incorporated and a fault tolerant design through extensive built-in-test has been introduced. These steps reduce our cost for operating and maintaining the 1000 aircraft fleet by nearly 20%. In the next step, where the Avionics Intermediate Shop is eliminated and everything else is kept the same as B, there is a further reduction in cost of over \$200M. Finally, introduction of standard modules and elimination of LRUs in favor of line replaceable standard modules, provides a further reduction in cost to approximately one-half of the current ownership bill. While these numbers aren't hard and fast they do provide an indication of the sizeable cost gains which can be realized through the application of PAVE PILLAR technologies.

TECHNOLOGY TRANSPARENCY

The building block approach espoused in this article permits not only the initial development of a highly flexible avionics suite, but also the continued development and integration of Pre-Planned Product Improvements (P³I). This is accomplished within the architecture by the concept of standard data buses (Parallel Interface (PI), Test and Maintenance (TM), High Speed Data Bus (HSDB)) and networks (SDDN, VDDN, DEN, Data Flow Network). This concept is implemented by Form, Fit, Function and Interface (P³I) specifications for each module type thereby permitting different designs by vendor for each module type and specific vendor module design modification dependent upon technological improvements.

Due to the open architecture, as new building blocks are identified they can be readily integrated into the avionics system thus permitting the performance upgrade to an existing aircraft at a relatively low cost. New investigations have already begun in the areas of parallel processing, artificial intelligence processing, and optical processing. The intent is to integrate these advanced processing elements into the PAVE PILLAR architecture if/when they become viable both operationally and logistically.

In summary, a PAVE PILLAR avionics architecture will result in dramatic improvements in availability, mission effectiveness, and cost of ownership. For reasons of mobility, logistics cost control, and austere maintenance of aircraft at forward sites, a two-level maintenance concept with line-replaceable-modules is endorsed by the PAVE PILLAR system. PAVE PILLAR plans to achieve these benefits and goals through advances in technology, and a systems approach to avionics integration. PAVE PILLAR is expected to provide the generic integration approach and architecture that will act as the foundation for avionics development in next-generation Air Force aircraft.

ADVANCED AVIONICS ARCHITECTURE

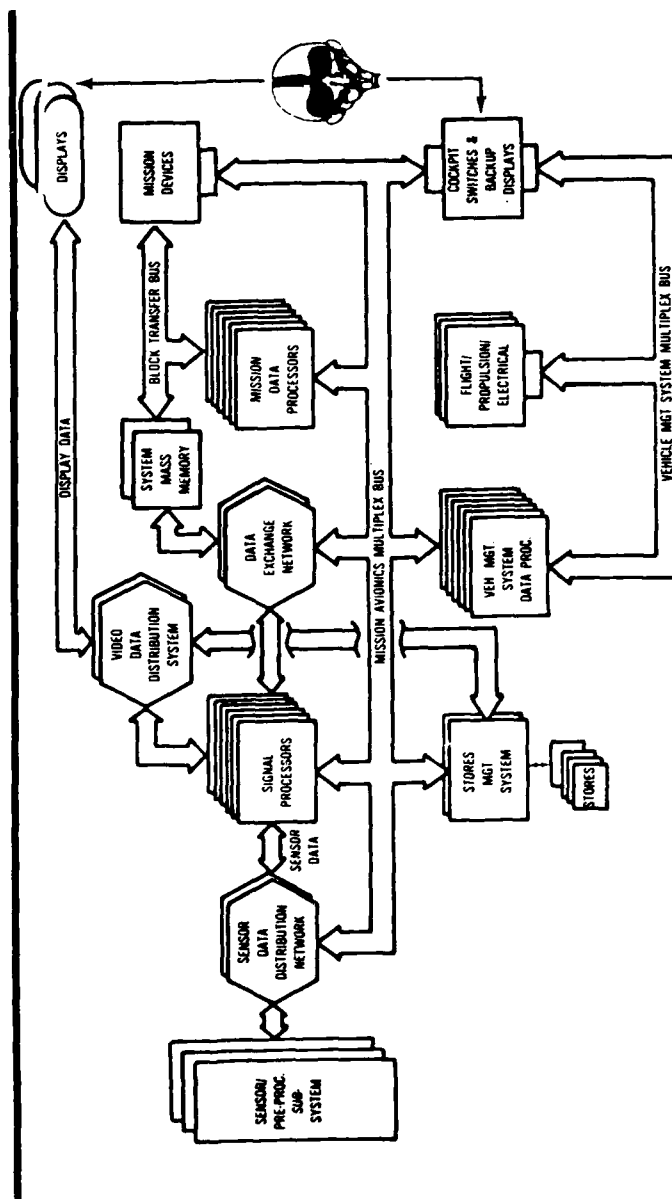


FIGURE 1

FUNCTIONAL PARTITIONING OF SYSTEM SOFTWARE

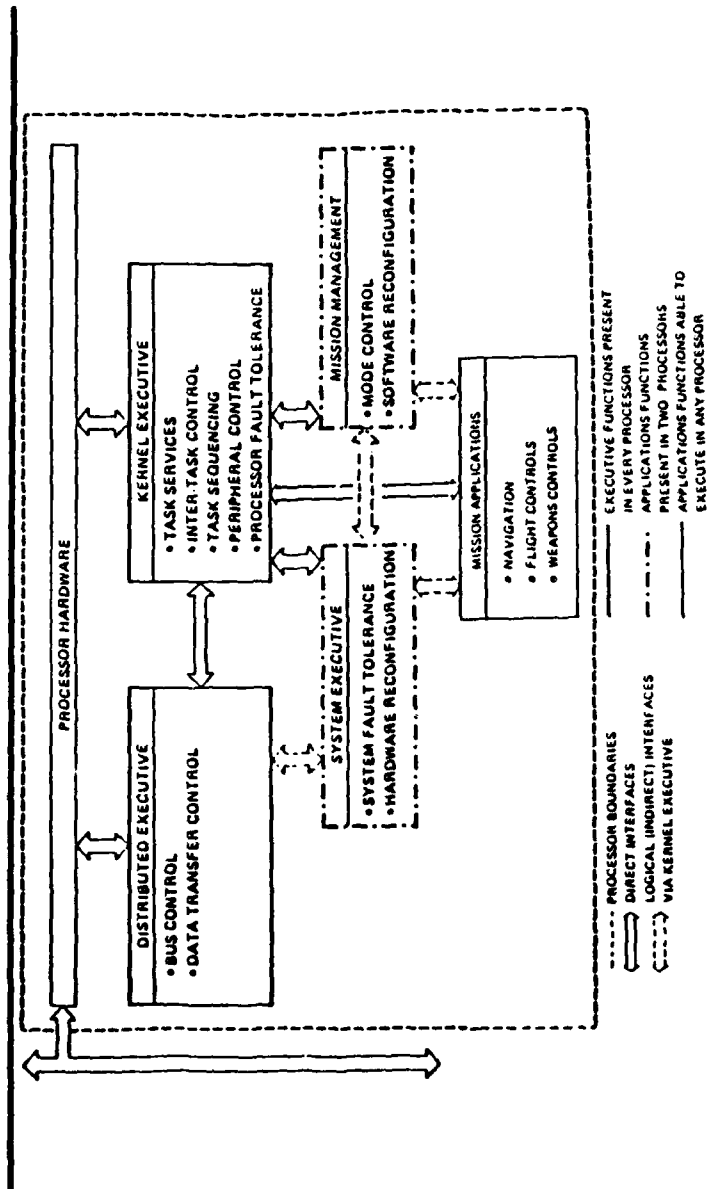


FIGURE 2

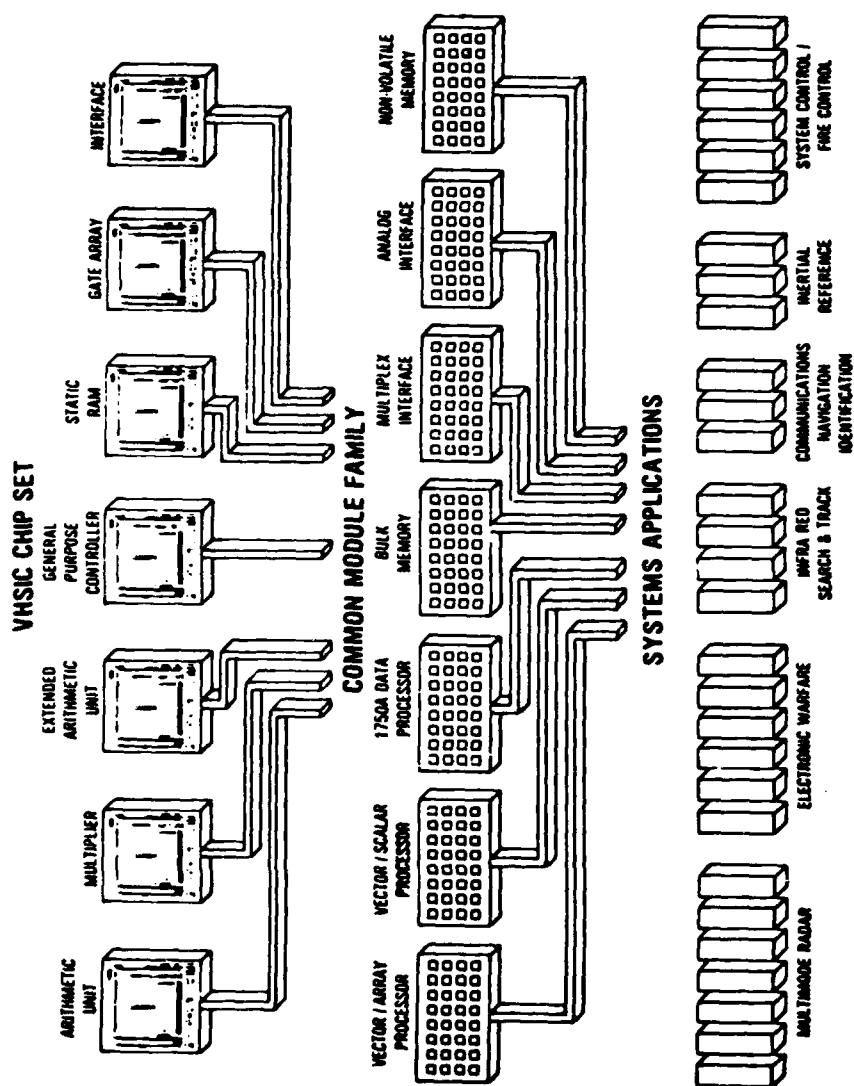


FIGURE 3

AVIONICS COST IMPACT OF MAINTENANCE CONCEPTS (PRELIMINARY RESULTS)

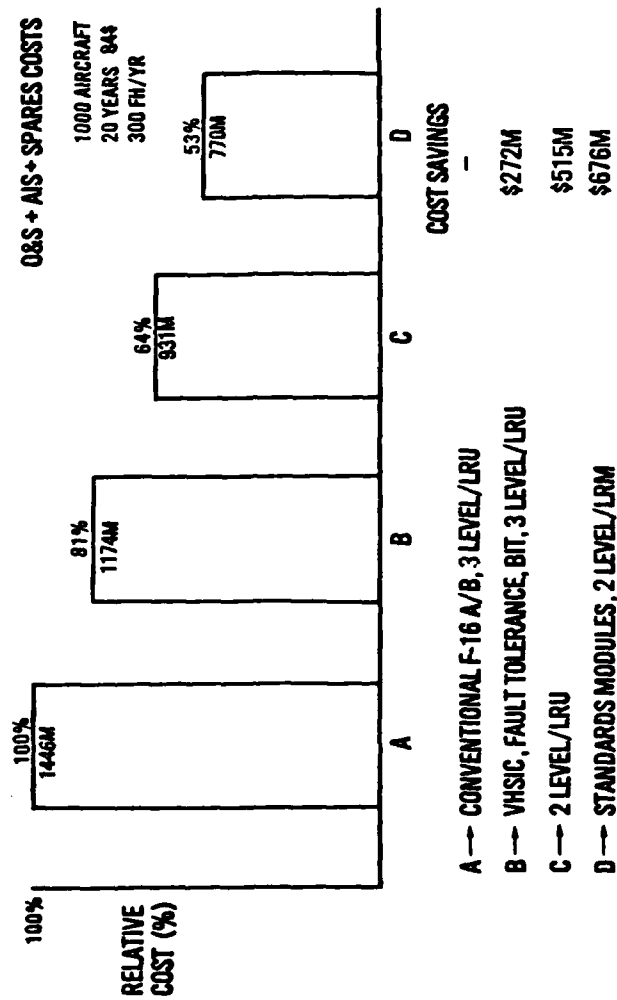


FIGURE 4

Data and Signal Processing Architectures for Future Avionics

Dr Mark T. Michael, Project Engineer
 Data & Signal Processing Group
 Air Force Wright Aeronautical Laboratories
 AFWAL/AAAT-2
 Wright Patterson AFB, Ohio 45433

Summary

Pave Pillar architecture incorporates a "Common Signal Processor (CSP)" concept as a key building block of a USAF advanced avionics suite. This CSP concept embodies the use of standard internal interfaces, a family of modules, use of the programming language Ada to express application program for the data processors within CSP, and the use of a Graph Notation to represent signal processing functions for the signal processing components of CSP. The modularity permits upgrading any hardware or software component with minimal disruption to the rest of the design. CSP is an "open architecture" in that the interfaces and module specifications are non-proprietary and can be built by other vendors. The paper will address the system concept, hardware architecture, and software philosophy comprising the CSP System. It will describe in general terms how the CSP hardware works and its expandability. Existing modules will be listed and potential future modules identified. A brief description of the Graph Notation and its advantages will be provided. The CSP Local Operating System capabilities and use will be summarized. Application studies to determine the suitability of the CSP concept for different avionic applications such as radar, electronic warfare, communications, and electro-optical sensors will be briefly summarized. The role of MIL-STD-1750A processors in CSP and their limitations will be touched on, and future upgrades to 32-bit machines discussed.

Background

AFWAL has been pursuing the development of generic digital signal processing architectures for avionics since the early 1980's. This is in response to a need to reduce the cost of signal processing subsystems, the time required to field them, and the difficulty encountered in maintaining and upgrading them. Most current signal processors are special purpose designs with little flexibility. Circuit technologies available in the 1970's did not permit highly programmable, modular designs within the size, power, and weight constraints of avionics. With the rapid advances in VLSI technology, order of magnitude increases in performance are possible compared with earlier circuitry. Hence, competitive performance common signal processor architectures have become feasible.

Several exploratory development in-house and contractual efforts were conducted which led up to the initiation of the advanced development CSP System project. In-house activities concentrated on evaluation of existing digital signal processors and identification of signal processing requirements across the avionic disciplines of radar, integrated communication/navigation, and electronic warfare. Parallel contract were let to AT&T Bell Labs and TRW to do additional requirements studies, design, and simulation of generic signal processing architectures. These efforts helped confirm belief in the feasibility of developing a widely-applicable common signal processor that would be performance competitive. The results of these efforts were used in the formulation of the requirements, baseline architecture, and performance goals for the CSP project. IBM Federal Systems Division, Manassas, VA, was awarded the CSP advanced development contract in late 1984. This effort is developing the architecture, hardware, software, and support equipment to demonstrate the CSP System and is the basis for much of the information in this paper.

The first CSP unit to be delivered will be incorporated into the Ultra-Reliable Radar (URR) as its principal processing system. The URR is an advanced avionics radar system under development by AFWAL and has a very demanding processing requirement. Hence it provides an excellent test for whether or not CSP can meet its performance goals. The initial radar capabilities to be developed are air-to-air modes. Studies were done by IBM on the CSP contract to assess the suitability of the CSP architecture for Communication, Navigation, and Identification of friend or foe (CNI), Electronic Warfare (EW), and Electro-Optical (EO) processing. These studies showed that the core architecture is a good match for their requirements; several new processing element module types were identified that would be desirable to develop to efficiently handle the signal processing functions in these additional areas. This will be discussed further later in the paper.

CSP System Concept

The CSP System can be viewed as a multiprocessor designed to permit the Processing Elements (PEs) to process at the highest utilization rate as practical for a given application. (The PEs perform the signal processing "number crunching.") To this end, a parallel Data Network (DN) provides high bandwidth point-to-point data paths for PEs to connect to Global Memories (GMs), Sensor Interfaces (SIs), and other PEs. (Figure 1 provides a block diagram of the CSP hardware architecture.) GMs provide a large storage area for holding sensor data until a PE is ready to process it and for holding intermediate computational results between processing steps. The GM may hold reduced data for later post processing by a data processor, a MIL-STD-1750A module, and transfer to other avionic system components over a system bus. The Element Supervisor Unit (ESU) is a 1750A data processor designed to efficiently control Functional Elements (FEs), which is the general name given for any PE, GM, or I/O Module connected to the network. By dedicating ESUs to pairs of FEs, it is possible to ensure that process control activities (I/O scheduling, task scheduling, etc) do not idle PEs or other FEs.

In order for CSP to be widely applicable with competitive performance and cost-effectiveness, it must be modular so that it can be configured for the processing needs of the application. The Data Network (DN) is the key to achieving this. It can be configured in a variety of ways (see the Data

Network Operations section). It must be possible to design new FEs that interface to the DN and ESU and use much of the existing software without re-work.

Since multiprocessors are inherently more difficult to program than uniprocessors, it was judged necessary to provide High Level Languages (HLLs) in which it could be programmed. Ada is provided for programming "Application Command Programs" (ACPs) within the 1750A Data Processors. A directed flow graph language, called Graph Notation, is provided for programming the signal processing portion of the system. It permits the programmer to describe parallel FE tasks conveniently. The CSP hardware and its resident Local Operating System (LOS) may be viewed as a "virtual multiprocessor" system that executes ACPs and signal processing graphs. ACPs control graphs through a set of LOS services. Off-line support software tools permit the preparation and simulation of ACPs and graphs prior to execution on the CSP hardware.

Reliability, maintainability, and availability received strong emphasis in the CSP project. VLSI circuits are designed with significant selftest capability. Modules include fault log devices for storing faults that may occur in operation. The system includes separate maintenance buses for recording faults and accessing them operationally and for diagnostic purposes. The system is designed for on-equipment maintenance in an operational setting; that is, sufficient testability is provided so that failed modules can be identified with minimal external equipment and replace.

Adequate technology transparency is an important goal for the CSP project. In order to avoid obsolescence, it must be possible to upgrade FEs and data processor with improved circuitry while retaining much of the existing software and hardware. By writing software in Ada and Graph Notation, significant hardware transparency is achieved for application software. FEs are defined with complete functions contained within the module, thus simplifying the interfaces and permitting performance and function evolution within each module without disrupting other system components. Maintaining an open architecture will permit other vendors to develop new modules to better meet their unique requirements than might be available within the generic set of modules.

Data Network Operation

The Data Network (DN) provides high speed parallel paths between pairs of Functional Elements (FEs) attached to it. It is designed so that a variety of configurations can be used. It is implemented with a VLSI circuit called the "Data Network Switch" (DNS), which is a switch with 8 16-bit ports. Each port has 4 control and parity lines associated with it. It permits any port to be switched to any other port. DNSs can be cascaded to form a multi-level network. Paths through the network are set up by routing information contained in the header of each message. Up to 4 levels of routing can be used with the circuit DNS circuit design.

The CSP Breadboard (BB) is configured with 8 Data Network Element (DNE) modules which provide 24 ports with 32-bit data paths. Each DNE uses 2 DNS circuits which are connected to provide 12 external 16-bit ports. The 16-bit ports can be viewed as a 16-bit slice and combined with another port to provide a 32-bit path. The second port can be on the same module or a second parallel module. Thus both 16-bit and 32-bit networks can be configured in a variety of ways. Figures 2 and 3 illustrate the DNE module layout and the CSP BB DN design, respectively. Table 1 lists some of the DN configurations that are possible with the DNS chip design and DNE module design.

The control of the DN is distributed in the sense that each message contains the routing information to determine its destination. There is no centralized control mechanism; this makes the design readily expandable. DNs with multiple levels of switching contain the possibility that messages may encounter busy nodes within the DN; that is, it is a blocking network. The alternative, a full crossbar switch which provides a direct path from every port to every other port, requires considerably more hardware and interconnections. It becomes impractical for more than a limited number of ports. If a message encounters a busy node, the design includes the mechanisms needed for retries. Software may choose to re-route the message over an alternate path. Initial use of CSP has shown that the DN has not proven to be a bottleneck for system operation.

The DN permits data to be transferred at one word per system clock once the path is established. A one-clock delay is added for each DN node through which the message must pass. Once a message has been initiated, path establishment through the DN is a hardware function requiring only a modest number of clock cycles to achieve.

Element Supervisor Unit (ESU)

The ESU module is a 1750A processor designed to efficiently control attached FEs and synchronize its activities with other ESUs. Two local buses are provided to connect to FEs: the Element Control Bus (ECB) and the Element Maintenance Bus (EMB). The ECB provides Direct Memory Access (DMA) to ESU local memory by DMA controllers on the attached FEs. This capability is used to efficiently fetch Data Network channel control blocks, FE processing control blocks (signal processing "Macro" control blocks for PEs, or read/write control blocks for I/O Modules), or make data transfers from FE local store to ESU local store. The EMB is a serial bus for accessing test and maintenance information within the FE and providing it to LOS resident within the ESU.

The ESU also connects to a system-wide linear control bus, the Parallel Internal Bus (PI-Bus), and to a system-wide serial test and maintenance bus, the TM-Bus. The PI-Bus is used to pass tokens from one ESU to another indicating events within the execution of ACPs and graphs, for transferring sensor control commands, passing computed graph results to data processors for post processing, or communicating with system I/O modules. The ESU PI-Bus interface logic includes extra hardware to make token passing very efficient.

Within the CSP Breadboard (BB), the ESU is used to "emulate" the VHSIC 1750A modules (see Figure

1), since these modules were not scheduled to be ready in time for the BB. Hence, all 1750As within the system are ESUs.

Floating Point Processing Element (FPPE)

The FPPE is a pipelined processor with 2 parallel floating point computational paths capable of performing 5 floating point operations/machine cycle burst rate or a complex floating point Fast Fourier Transform (FFT) butterfly operation once every 2 cycles. As an FE, it is controlled by an ESU over the local ECB and EMB interfaces and is connected to the DN for data transfers. It contains local memory divided into 2 logical banks that operate as "swing" or "ping pong" buffers with one bank connected to the network and the other to the FPPE data flow logic. When a task completes, the banks are switched. Each bank holds 16K words of storage.

The FPPE contains microstore for holding its microprograms, referred to as "Macros." Local sequencers control Macro execution. Each Macro execution requires a "Macro Control Block" (MCB) to specify its starting address and a set of execution parameters. These MCBs are stored in ESU memory and accessed by the FPPE over the ECB using DMA transfers. A string of Macros comprise a task. The ESU initiates a task with an I/O transfer to the FPPE providing the sequencer with the starting address of a MCB string in ESU local memory. When the task is completed, the ESU is notified by the FPPE.

The FPPE is designed to permit continuous processing to be maintained. While the FPPE is processing one task using data in one bank of local store, the controller is unloading results from the previous task out of the other bank of memory and then loading input data for the next task. If this can be done before the current task completes, the FPPE will remain busy.

The FPPE's primary data type is 32-bit floating point with a 24-bit fractional part and an 8-bit exponent. It also operates on extended precision floating point with a 47-bit fraction and 8-bit exponent and 16-bit fixed point data types. Basic operations include add, subtract, multiply, compare, logical OR, AND, XOR, data dependent branches, sine, cosine, square root, arctangent. A lookup table facility is included to implement complex functions. A separate coefficient memory is provided for storing constants; this is useful for many signal processing computations and complex functions.

Global Memory (GM)

The GM provides a large memory storage element for holding digitized sensor data, intermediate computational results for signal processing PEs, and final reduced data for later transfer to data processors for post processing. It also is used to store program loads, code and initial data, for all of the computational units. Use of global memory provides a way to decouple data arrival from succeeding processing steps and to decouple those processing steps from each other; in this sense, it differs from "hardwired" signal processors with a "flow-through" sort of design that passes the data from element to element. The high bandwidth DN makes this concept workable.

The GM is an intelligent memory in that it provides six hardware-supported address modes for accessing data in it. These are

- (1) Single Circular Queue
- (2) Multiple Circular Queue
- (3) Corner Turn Queue
- (4) Coordinate Rotation
- (5) Random Access Queue
- (6) Buffer

and they provide the user with the ability to efficiently store and retrieve data in the ways needed for signal processing applications. Two dimensional arrays of data can be manipulated efficiently, such as encountered in "Synthetic Aperture Radar" (SAR) applications, for instance. The ability to define a variety of "Storage Objects" is provided in the Graph Notation, and the CSP System (LOS and the hardware) performs the operation for the user, relieving him of specifying the details of the operation.

The current CSP GM provides 1 million 32-bit words of storage. The addressing logic permits expansion to 4 million words. The GM has three ports: the DN, Element Control Bus (ECB), and the Element Maintenance Bus (EMB). The GM bandwidth matches the DN port bandwidth of 1 32-bit word/system clock. The memory includes an 8-bit Error Correcting Code (ECC) that corrects any errors in any one 4-bit nibble within the 32-bit data word. A logical to physical mapping mechanism is provided to support dynamic memory usage. A fixed size paging scheme is used to avoid memory fragmentation. The memory is organized into 4 banks of 256K x 40 bits which are accessed in an interleaved fashion; this permits slower memory circuits to be used and still match the DN bandwidth.

Sensor Interface (SI)

The SI module is a high speed port for connection to a sensor in a point-to-point manner or it may be used in a Ring network configuration connecting a series of CSPs together. The current implementation is a wire version with separate 32-bit data paths for transmit and receive (2 bits of parity are provided for each channel also). Consideration was given in the design for eventual upgrade to fiber optics. Data is transferred at a maximum rate of one word every 2 machine clock cycles. In full duplex mode, one channel may operate continuously while the other operates in a pulsed manner.

The SI Module is controlled by an ESU over the ECB and EMB interfaces. It connects to the Data Network, which operates at a rate of one word per machine cycle. The SI contains 3 banks of local memory which may be switched from one interface to another when they are filled or unloaded. Transfers may take place from ESU local memory to the SI local memory and vice versa; from the Data Network to

and from the SI local memory; and from SI local memory to and from the external SI interface.

A minimal message protocol is available for when the SI is in point-to-point Link mode; a Ring protocol is available for operation in the Ring Network mode. Messages consist of frames consisting of several header words, from 0 to 4K data words, and several trailer words. ESU management of the SI is similar to the other FEs: message control blocks are stored in ESU program store and are accessed over the ECB by the SI controller logic. Each message requires a control block. A string of control blocks may be processed by the SI without ESU intervention; this is an SI "task" to the graph programmer. The ESU initiates an SI task with an I/O command providing information to the SI about the control block string.

System I/O Modules

The CSP BB includes a MIL-STD-1553B I/O Module for communication with other avionic equipment having that bus port available. A High Speed Data Bus (HSDB) I/O Module will provide system communication in the future.

Support Modules

In order to complete the CSP BB, various support modules must be included. A Timing and Control Generator (TCG) provides the system clocks needed for the various FEs and system buses. A User Console Interface (UCIF) provides a laboratory connection to the User Console (UC); it permits the UC to control the CSP System. An IEEE-488 bus connects the UC to the UCIF; the UCIF can be viewed as a gateway to the PI-Bus and TM-Bus within CSP. The TM-Bus permits it to control any ESU. The PI-Bus permits the UC to efficiently transfer data to any ESU or selectively monitor PI-Bus activity.

Distributed DC to DC power convertor modules are used to supply power to the logic modules. A laboratory source is used to supply the intermediate DC voltage to the convertor modules. Two modules are included for PI-Bus and TM-Bus terminations.

Graph Notation

A CSP Graph is a high level method for representing the signal processing functions executed within CSP using data flow techniques. It is one of two basic program units used to program the machine, the other being an Ada Application Command Program (ACP). A "node" (task) within a graph "fires" (executes) when all of its inputs are available and its output buffers are empty. A task does not retain data; data is passed through the graph links to the next node in data flow fashion. Graphs start executing after an ACP has "instantiated" the graph (linked and loaded the graph from a "graph realization" stored in a GM to form a "graph instance"), enabled it, and input data is available.

An ACP controls a graph through a set of Local Operating System (LOS) Services. An ACP may provide Graph Constants as parameters for a graph instantiation to specialize it for a particular function. Graph Variables may be supplied to an executing graph for changing its parameterization; Graph Switches may disable or enable nodes in the graph. An ACP may connect one instance of a graph to another instance, or delete a previously created instance. ACP tasks may be connected to CSP Graphs through graph interfaces.

A CSP Graph provides a convenient notation for specifying tasks for CSP Functional Elements (FEs) that relieve the programmer from having to specify low level hardware operations yet use the full capabilities of the various FEs. Two types of GM Storage Objects, Queue and Buffer, may be specified that permit the full range of GM addressing modes to be used to access data. FPPE tasks consist of Macro (primitive) strings that perform signal processing functions on a set of input data loaded into FPPE local store and unloaded when the task is finished. Graph interface tasks may be specified for transferring messages on the CSP data paths (the Data Network, PI-Bus, and Sensor Interface). A sample graph for a Spotlight Synthetic Aperture Radar (SAR) function is provided in Figure 4.

Local Operating System (LOS)

LOS is the resident real-time operating system within the 1750A data processors that controls the CSP System and "runs" the ACPs and Graphs. LOS Services provide the interface through which the ACP commands the sensor, interacts with other Pave Pillar architecture subsystems, communicates with other ACPs, and controls the graphs. LOS includes an "Availability Management System (AMS)" that monitors hardware and software error reports, filters these reports, stores them locally and in non-volatile module Fault Log devices, and takes corrective actions. It includes system startup capability and debug features for development purposes. It includes system resource management functions for controlling what is a distributed "virtual multiprocessor." It controls the logical to physical mappings for management of the Global Memories (GMs).

LOS is divided into three components: the Subsystem Master (SSM) LOS resident in the controlling 1750A data processor, the SSM Support Processor (SSMSP) resident in a secondary 1750A, and ESU LOS resident in each ESU controlling FEs. ACPs are resident in SSM and SSMSP 1750As; ESUs only contain LOS code and data. ESU LOS, a single program, is capable of controlling any FE type: FPPE, GM, or SI. The critical function of ESU LOS is to schedule, dispatch, and run graph tasks efficiently in order to avoid idle FEs. This means that PI-Bus messages, Data Network messages, and ECB message transfers must be handled efficiently. LOS essentially implements the graphs as sets of data tables and control block strings defining the graph functions to LOS and the attached FEs. Graph instantiation consists of the loading of these tables and strings in ESUs appropriately. Task scheduling is done with a set of prioritized First-In-First-Out (FIFO) queues; task dispatching is done with a single non-interruptible FIFO queue.

Support Software Environment (SSE) and Support Equipment

Figure 5 provides a block diagram of the support tools comprising the CSP SSE. The SSE is hosted on a Digital Equipment Corp (DEC) VAX computer system. The CSP is directly controlled by a PC AT User Console (UC) in a laboratory environment. Software is downloaded from the VAX through the UC to the CSP. The SSE is a set of tools for preparing and simulating (1) Ada Application Command Programs (ACPs), (2) signal processing Graphs, and (3) FPPE microprograms.

An Ada/1750A Language System is used to prepare ACPs. It consists of a compiler, Macro Assembler, Linkage Editor, PI-Bus-based multiple computer simulator, and an Ada Debugger. The Debugger is being extended to provide symbolic access to the CSP hardware; it currently is usable only with the simulator.

A Graph Translator translates Graph Notation, written in textual form, into a form usable by the System Level Simulator (SLS) and a form usable by LOS for loading into the hardware. The SLS simulates the control flow of the execution of a CSP Graph. It provides loadings on CSP resources: FEs and all data paths, and thus is useful for analyzing a graph design to see if it is using the CSP resources effectively. It does not execute any data; rather, it is a discrete event simulation that models the execution times, message passing, memory usages using estimates derived from FPPE microprogram simulations for FPPE tasks, built in LOS overhead times, GM, SI, PI-Bus, and DN performance characteristics. The simulation is maintained to reflect an accurate model of the actual CSP System performance.

FPPE tasks are written in a microprogram language, assembled with a Microassembler, and simulated on a Microsimulator. The Microsimulator simulates the microprograms ("Macros") at the machine code level with a clock-level accuracy, thus providing exact execution time estimates and computational results. Strings of Macros, graph tasks, may be simulated to check out complete graph task nodes. A library of Macros is maintained which may be re-used in various applications. A System Build program generates a load image for the CSP System which includes the required Macros as well as the 1750A Ada and assembler load images for eventual loading in CSP through the User Console.

A Support Equipment suite is provided which consists of a mini-CSP that can be used to simulate SI sensor inputs, 1553B avionic system inter- actions, and accept CSP outputs. It consists of a UCIF, several ESUs, a GM, Lwo Sis, and a small Data Network. It is controlled by the User Console. Since it is a CSP, the same support software tools are used to prepare its software as that for CSP.

Application Studies

Requirements and design studies were done for assessing the suitability of using CSP in CNI, EW, and EO signal processing subsystems. Any changes to the core architecture components and addition of new modules were identified. Table 2 lists the core architecture modules, support modules, and a set of candidate new modules. The Vector Signal Processing Element (VSPE), General Purpose PE (GPPE), Sort Enhanced PE (SEPE), and the Bi-phase Correlator PE (BCPE) were defined and preliminary designs developed during the application studies.

The VSPE is a PE that processes 16-bit fixed point data. It is a pipelined processor with microprogrammed control and standard FE interfaces: DN, ECB, and EMB. It is similar in architecture to the FPPE, but somewhat less flexible. It will work well on certain types of well- structured vector or matrix-oriented algorithms, including FFTs. For these algorithms, it provides a times four improvement in throughput over the FPPE.

The GPPE is a data processor attached to the Data Network and control- led by an ESU using graph control procedures. It is targeted for decision-intensive signal processing functions often found in EW and some CNI applications. IBM specified it as a 1750A processor, but a Reduced Instruction Set Computer (RISC) class machine may be a better match for the requirements. A CSP configured with all GPPEs can be viewed as a general purpose parallel processor using both Ada and Graph notation as its programming languages.

The SEPE is designed for performing EW data sorting functions at very high speed. These functions are often performed in special purpose hard- wired units today. Associative comparators augment the arithmetic logic. It is suitable for filtering long pulse descriptor words created by sensor front-end units into clusters for later post processing. It creates hit counts on bins and thresholds these. Up to four compare parameters may be specified per bin. Inputs are sorted into time-of-arrival order within each bin. Input word lengths may be 16, 32, 64, or 128 bits. It conforms to the FE standard hardware and software interfaces.

The BCPE was designed to perform communication-oriented signal processing functions such as found in match filter algorithms, preamble detection algorithms, and other code processing functions. It conforms to the standard FE hardware and software interfaces.

Advances in Data Processor Architectures

CSP uses MIL-STD-1750A processors to perform the data processing and local control functions. This USAF standard is a 16-bit Instruction Set Architecture (ISA) with 16 16-bit general registers and direct address- ability to 64K words of instructions and 64K words of data. It is a moderately complex ISA with over 200 instructions and a dozen address modes. To address more than 64K words, a page mapping option must be implemented. All implementations must include floating point.

Within the past 10 years, it has been demonstrated that "Reduced Instruction Set Computers" (RISCs) provide a more cost-effective use of circuitry than the "Complex Instruction Set Computers" (CISCs). Initial demonstration of this principle was done at IBM on a project known as 801; work at the University of California Berkeley in the early '80's continued the effort, adding some interesting

register usage concepts to the design, and also coining the term RISC. RISC designs are based on the principle that only a small set of basic operations form the bulk of instructions most computer programs execute and that the instruction set should be restricted to those basic instructions. More complex operations can be synthesized from the simpler ones. Historically, the development of CISCs can be partially attributed to the need to use slow, expensive core memories. CPUs were microcoded; adding more instructions and address modes was inexpensive once the microengine was designed. And they provided marginal improvements in program size and execution times. With solid state memories more nearly matching logic speeds and continually increasing in density, the CISCs are no longer cost-effective.

RISC architectures are an excellent match for a variety of avionic data processing requirements. Many avionic applications are throughput intensive rather than memory intensive. RISC take less hardware for a given throughput. Addressing large memories directly is straightforward with 32-bit machines, simplifying the software for those applications needing large memories.

DOD has sponsored the several RISC projects, notably the Defense Advanced Research Projects Agency (DARPA) Microprocessor without Interlocked Pipeline Stages (MIPS) projects. These projects are developing both silicon VLSI and Gallium Arsenide (GaAs) VLSI MIPS microprocessors. The MIPS architecture is based on work done at Stanford University. AFMIL has initiated exploratory efforts to assess the feasibility of combining selected stack machine features with RISC principles to achieve a more High Level Language (HLL) compatible 32-bit RISC design.

The proposed concept of a common 32-bit data processor architecture is to develop an ISA with a limited number of subsets which would permit the machines to be tailored to the application. A RISC core set of instructions would be common to all machines. Baseline additional sets include floating point, supervisor/task support, memory protection, and memory mapping. Good multiprocessing support is a requirement. The ability to use cache memories effectively in the machine designs is necessary for larger applications.

Within integrated signal and data processing architectures such as the CSP System, such a 32-bit processor could find a variety of applications. It could serve as a Functional Element (FE) processing decision-intensive signal processing functions often found in EW and CNI. Such a module would be designed to meet the CSP interfaces and operate under the control of the existing LOS. It could serve as a post processor in place of the 1750A modules. It could eventually replace the 1750A engine within the ESU module, although this will require re-targeting the LOS to a new ISA.

Some preliminary studies indicate that RISC processors are a good match for Artificial Intelligence (AI) applications. If this is so, that may aid the introduction of those techniques into real-time avionic applications by providing a significantly higher level of throughput than currently available.

An important development area for 32-bit RISC processors for avionics is in multiprocessor systems using software-transparent common memory. Within the context of a CSP System, the functional group (ESU controlling PEs, GMs, or I/O Modules) may evolve to a form of shared memory multiprocessor with the attached modules serving as peripheral processors. As VLSI designs advance, the RISC processor can constitute a circuit "macrocell"; signal processing functions will be other macrocells on the same circuit. RISC macrocells, cache memory macrocells, pipeline ALUs macrocells, etc. could comprise a small library of macrocells from which different parallel processor "nodes" could be constructed. Use of a microcircuit technology such as Gallium Arsenide (GaAs) will permit a very fast RISC processor PE to be built when that technology is mature enough to support VLSI digital circuits.

Some Lessons Learned To Date on the CSP Project

All DOD computer systems developed after January 1984 are required to use the Ada programming language. Since CSP was started in November 1987, it falls under that mandate. CSP is one of the first major avionic systems to use Ada for large amounts of software. Hence, in a sense, it is a test case for Ada. Much, but not all, CSP software has been written in Ada. Some of Ada's good points are that the code is more readable than most languages, the programmers tend to write better-structured code, debug times are less than for older languages (the extra constraint checking helps to find more problems at compile time), and it is especially well-suited for large software systems written by a team of programmers. The design time tends to be longer with Ada, but the testing time is shorter, often only half what it normally takes. The language is wordy in that it typically requires almost twice the number of Source Lines Of Code (SLOC) that other languages take, but this extra code is mostly compile time information that does not produce machine code. The code efficiency of the selected Ada compiler for the embedded 1750A controllers is about the same as that for good JOVIAL J73 compilers for the "sequential" (non-tasking) portion of the language. This assumes that the constraint checking built into the language (range, index, type checking) has been suppressed once the code has been debugged. Constraint checking may add as much as 100% to the code size and run-time or as little as 20% depending on the nature of the code. Ada seems best suited for application programs and least-suited for operating systems or control programs that must closely manage the computer resources. The DOD has tightly controlled the language definition and compiler certification process compared to other languages, and in the long run, this should be a great benefit for achieving commonality and software portability.

An obvious difficulty, especially at the start of the project, in using Ada was the immaturity of the available tools. It has taken a long time to achieve usable Ada compiler systems. This is due partly to its newness, but also to the complexity of the language and its novel features. The certification process, despite its extensiveness, has not guaranteed that the compilers will produce correct code; it tests nothing with respect to efficiency. Fortunately, the CSP project selected probably the best Ada compilers available at that time and so the negative impact on the project has been minimized. (Three Ada compilers are in use on the project: the DEC VAX compiler being used for VAX-hosted support software, the Alsys PC AT compiler being used for User Console software, and the TLD

Systems, LTD VAX-hosted cross-compiler being used for the embedded 1750A controllers. Substantial resources are being spent to improve the TLD toolset for CSP use.) The CSP project began with the intention of using Ada to code the great bulk of the Local Operating System (LOS). Even the Ada parallel processing (tasking) features were going to be used for the internal design of LOS. But, unfortunately, the Ada tasking model is a poor match for message-based distributed processing systems. Too many rendezvous (Ada's basic task synchronization feature) were needed to perform key functions which could be done with half as many simpler primitives. The rendezvous itself is quite complex and inefficient. (The rendezvous can undoubtedly be made more efficient as compilers are optimized, but there is an inherent complexity in it that cannot be eliminated). Ada tends to remove the programmer from the machine resources, but since an OS manages those resources, use of Ada tends to prevent the programmer from accomplishing his task. Following an evaluation of Ada tasking, the LOS was re-designed to use conventional tasking primitives written in MIL-STD-1750A assembly language. "Sequential" Ada is used to code portions of LOS. Application Command Programs (ACPs) are being written in Ada and these programs may use Ada parallel processing, although it is discouraged due to its inefficiency. The scope of an Ada program is restricted to a single 1750A processor; communication between processors is provided by LOS services.

The CSP project has made extensive use of chip, module, and unit level simulations in order to test the hardware designs. Even more simulation use would have been beneficial. Initially, only pure software simulations were done. These are practical for single chip logic simulations and some multi-chip simulations, but they are too slow for adequate module and unit simulations at the logic (gate) level. The largest mainframe computer cannot provide reasonable turn-around times for these more extensive simulations. Unfortunately, design errors often exist at the interfaces between components: chip-to-chip or module-to-module, since they are usually designed by different people. To provide more simulation throughput, IBM acquired a logic simulation accelerator from ZYCAD Corp. This unit can be attached to a host computer and used as a peripheral processor. It provides several orders of magnitude improvement in typical logic simulation times. It has arrays of programmable logic elements that can be programmed to model the circuit at the gate level (modelling at the transistor level is also possible). The logic elements operate in parallel at TTL-class speeds when the simulation is run. Results are returned to the host computer. The user can monitor his circuit operation very conveniently. Use of this tool significantly increased the amount of circuit testing that could be done before release of chips and modules to fabrication. Significant portions of the operating system were run on the simulation before the hardware was available. Development of complex "Application Specific ICs" (ASICs) for use in highly integrated systems in a timely manner requires the use of such a simulation tool in the opinion of the author.

CSP can be viewed as the development of a "coarse-grained" parallel processor in that a moderately large number of processing elements and memory elements can be connected to its Data Network and operated in parallel. In order to program such a system in a reasonable length of time, the user needs support software tools to make his job simpler than was the case for previous programmable signal processors. Development of these tools is expensive and time-consuming, but necessary. This was recognized at the beginning of the project and planned for. A rough estimate is that CSP will include the development of over a half million SLOC before the project is completed. Similarly, the development of the hardware was expensive. A dozen large semi-custom ICs and a half dozen gate arrays were developed in order to complete the core design. Considerable effort was made to keep the ICs as generic as possible and the number to a minimum, but how to do this can often be best seen after the design is done. It seems clear from the CSP experience, that in order for parallel processors to be affordable, they must be widely applicable and they must be relatively technology transparent in order to amortize the development costs across a reasonable set of applications. It is felt that the CSP System can meet these goals, but that needs to be demonstrated in actual applications. Perhaps as more is learned about how to design parallel processing architectures, designs will become simpler and less expensive, but the CSP experience suggests that useful parallel processors will remain very expensive to develop compared to serial processors for the immediate future.

Data Network Element (DNE) Modules Required

| #DN Ports | 16-bit Paths | 32-bit Paths |
|-----------|--------------|--------------|
| 6 | 1 | 1 |
| 8 | 1 | 2 |
| 12 | 2 | 4 |
| 18 | 3 | 6 |
| 24 | 4 | 8 |
| 30 | 5 | 10 |
| . | . | . |

Table 1. Data Network (DN) Configurations

Core Architecture Modules

DNE - Data Network Element
 ESU - Element Supervisor Unit
 FPPE - Floating Point Processing Element
 GM - Global Memory
 SI - Sensor
 V1750A - MIL-STD-1750A Data Processor, emulated with ESUs]
 1553B - MIL-STD-1553B Input/Output Module

Support Modules

TCG - Timing and Control Generator
 UCIF - User Console Interface
 IOT - I/O Terminator for the PI-Bus
 PSs - Distributed DC-DC power convertors

Candidate Additional Modules

VSPE - Fixed Point Vector Signal Processing Element
 GPPE - General Purpose Processing Element
 SEPE - Sort Enhanced Processing Element
 BCPE - Bi-phase Correlator Processing Element
 HSDB - High Speed fiber optic Data Bus Interface
 GAPE - Gallium Arsenide Processing Element
 FOSI - Fiber Optic Sensor Interface

Table 2. CSP Modules

References

1. Common Signal Processor (CSP) Principles of Operation, Contract No. F33615-84-C-1470, CDRL 1-12.10, 22 Jan 86.
2. CSP Support Software Users Manual - Graph Notation Reference Manual, Contract No. F33615-84-C-1470, CDRL 1-15.01, 11 Dec 86.
3. CSP Interface Control Document (ICD) for Application Software to LOS, Contract No. F33615-84-C-1470, CDRL 1-33, 26 Mar 86.
4. Generic Signal Processor Architecture (GSPA) Final Report, AFWAL-TR-86- 1017, Vols. I-III, Feb 86.
5. MIL-STD-1750A (USAF), Sixteen-bit Computer Instruction Set Architecture, 2 Jul 80.
6. MIL-STD-1553B (USDOD), Aircraft Internal Time Division Command/Response Multiplex Data Bus, 30 April 75.
7. MIL-STD-1815A (USDOD), Reference Manual for the Ada Programming Language, Jan 83.
8. R.O. Simpson, P.D. Hester, "The IBM RT PC ROMP processor and memory management unit architecture," IBM Systems Journal, Vol. 26, No. 4, 1987, pp. 346-359.
9. D.A. Patterson, C.H. Sequin, "RISC I: A Reduced Instruction Set VLSI Computer," manuscript of talk for Eighth Annual Symposium on Computer Architecture, UC Berkeley, CA, May 81.

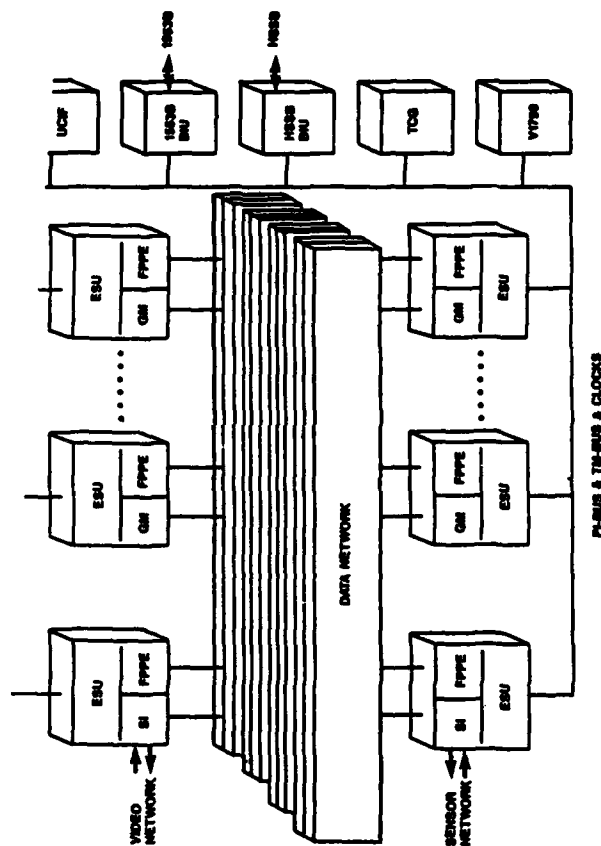


Figure 1. Common Signal Processor (CSP) Hardware Architecture

8000 1000

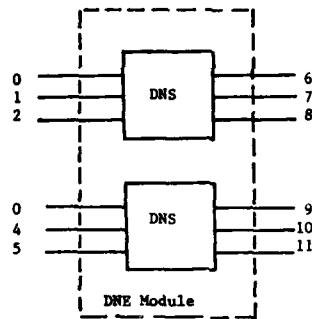


Figure 2. Data Network Element (DNE) Module and Data Network Switch (DNS) Chip

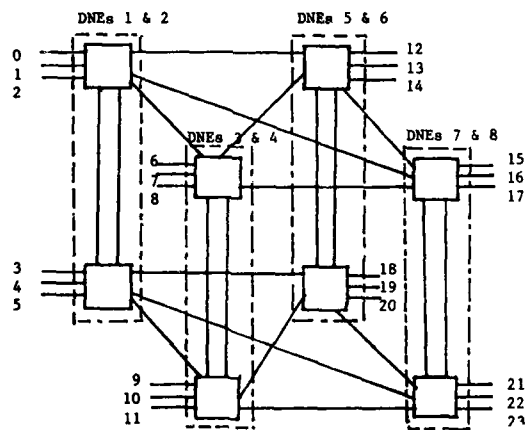


Figure 3. CSP Breadboard Data Network Configuration

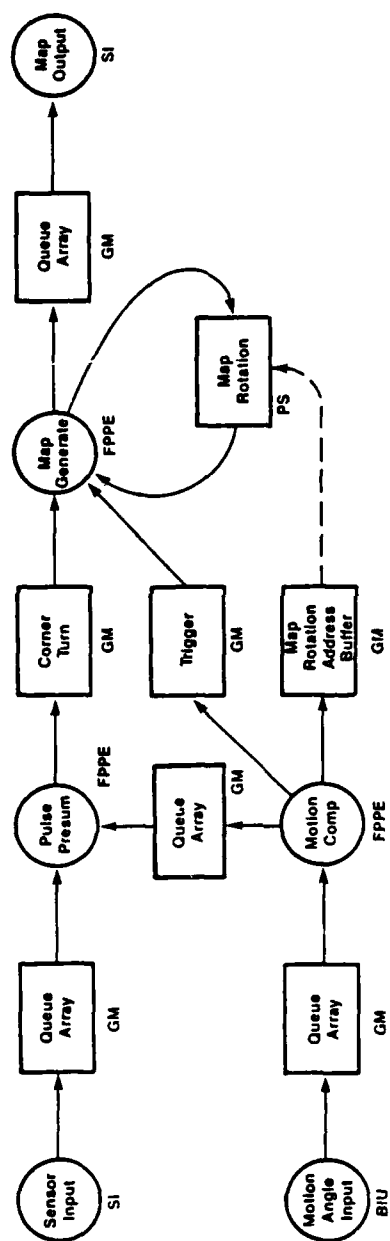


Figure 4. Spotlight SAR Data Flow Graph

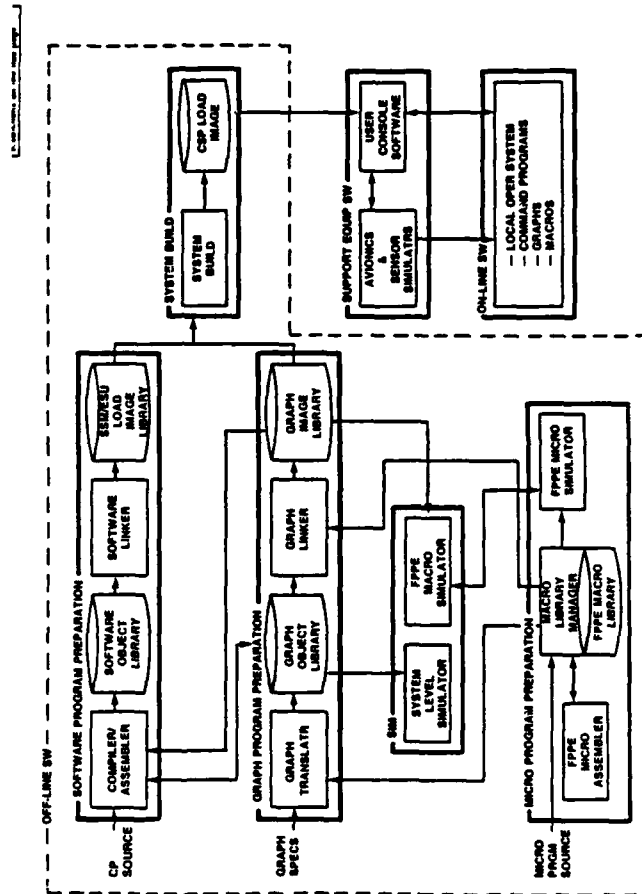


Figure 5. Common Signal Processor (CSP) Support Software Environment (SSE)

6-12-1988

FIBER OPTIC BUSES AND NETWORKS FOR ADVANCED AVIONICS ARCHITECTURES

by
R. W. Uhlhorn
Senior Principal Engineer
Harris Corporation, Government Aerospace Systems Division
P.O. Box 94000
Melbourne, Florida 32902
U.S.A.

SUMMARY

An avionics architecture exploiting high speed, high density VLSI and VHSIC technology by repartitioning the traditional avionics suite requires subsystem interconnection via backbone and backplane buses and networks operating at data rates far exceeding those used for avionics in the past. In fact, the rates are high enough that fiber optics is the only interconnect technology that does not impose substantial size, weight, and life cycle cost penalties on the overall system. In the Pave Pillar architecture the fiber optic multiplex bus for command and control, block transfer, and flight control functions, based on a variation of the IEEE 802.4 and 802.2 token-passing bus protocol for the physical and data link layers, operates at 50 Mbps. A number of specific implementations have emerged. The parallel internal (PI) bus protocol ties users (data processors) together in the backplane and through a bus interface unit to the multiplex bus. Other kinds of networks are used to serve subsystems connecting video terminals, sensors, signal processors, etc.

In this paper, an overview of the multiplexed high-speed data bus and parallel internal backplane bus designs and their interface is presented. The implementation details and options for the fiber optic network which supports the interconnection of avionics bus interface modules in different physical locations are discussed. Passive and active star-coupled networks are compared and conclusions drawn. The state of the art in packaging of the avionics bus interface and related line replaceable modules is illustrated.

1. INTRODUCTION

An advanced avionics architecture gaining widespread acceptance among designers of future aircraft, and finding applications in spacecraft and ships, evolved out of study programs sponsored by the U.S. Air Force. The Pave Pillar architecture, shown in Figure 1, capitalizes on the latest advances in electronic technology, such as Very Large Scale Integration (VLSI) and Very High Speed Integrated Circuits (VHSIC), and incorporates common elements among subsystems thereby offering life cycle cost benefits and unprecedented flexibility[1]. A new partitioning philosophy of loosely-coupled distributed processors is required to achieve these goals forcing subsystem elements to be interconnected with high speed, wide bandwidth data links. Modularity of the avionics components naturally follows. Distributed data bases add to the overall processing flexibility. Furthermore, the parallel data and signal processor interconnects must operate at high rates to keep up with the enormous processing capacity now possible.

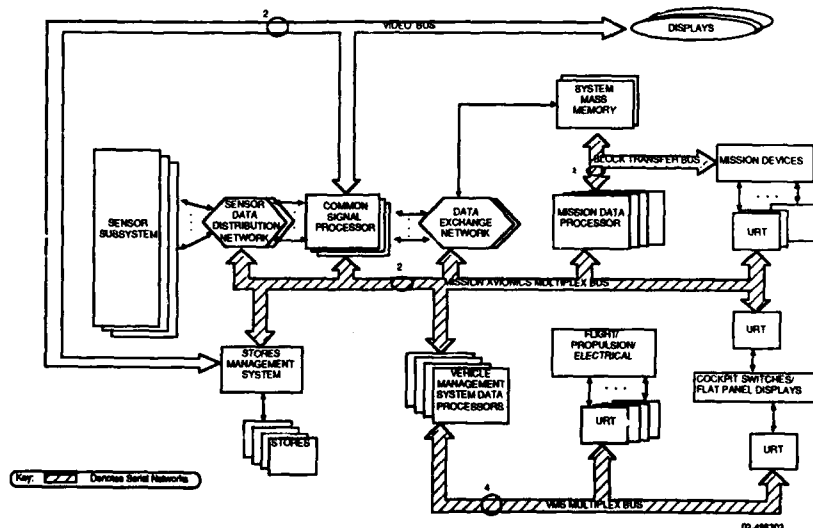


Figure 1. Serial High-Speed Data Bus Applications in the Pave Pillar Architecture.

In the architecture shown in Figure 1, the mission avionics multiplex, block transfer, and vehicle management buses all use common components (to a certain point) and operate at a serial data rate of 50 Megabits per second (Mbps). The sensor area network must have a throughput approaching 1 Gigabit per second (Gbps). The video bus, when implemented in a wideband analog format, requires a bandwidth of nearly 1 Gigahertz (GHz). From a size, weight, and life cycle cost standpoint, it can readily be shown that a

2. THE AVIONICS BUS INTERFACE ARCHITECTURE

3. THE HIGH SPEED DATA BUS PROTOCOL.

[illegible]

A brief discussion of the SAE data bus and protocol follows[4]. It consists of a set of stations broadcast-connected on the transmission medium—that is, each station which transmits is heard by all the other stations. A transmission is accepted based on physical or logical addressing mechanisms. Access to the fiber optic transmission medium is controlled by a token which is continually passed around a logical ring formed out of all the stations. A station receiving the token gains the right to use the transmission medium for a certain amount of time. The amount depends upon the value of the token holding timer (THT), used for all messages of priority 0 (high), as well as the residual value of the token rotation timers (TRTs), one for each decreasing priority from 1 to 3. The amount of time is always less than or equal to a predetermined maximum value based on circumstances of the bus traffic. When this amount of time has expired, or the station has sent all of its messages, then the station forwards the token to the next member of the logical ring.

Low latency for high priority messages is assured by the use of message priority TRTs. A station which has a message at the highest priority (0) always sends that message when it receives the token. A station which has a lower priority message sends it if the token rotation timer associated with that priority level has not expired. Otherwise, the station must forward the token to its successor. In this way the token-passing bus users defer to higher priority traffic when the traffic load becomes heavy.

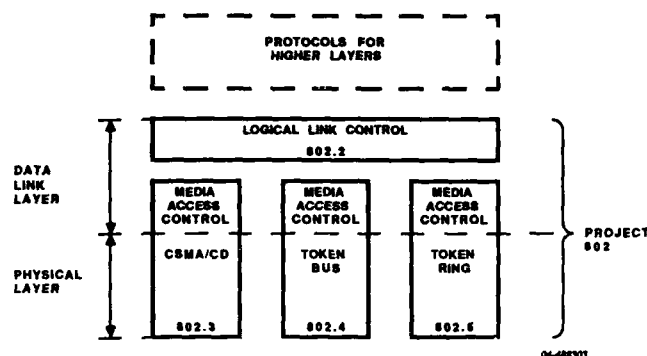


Figure 3. IEEE 802 Standardization Project Standards Structure.

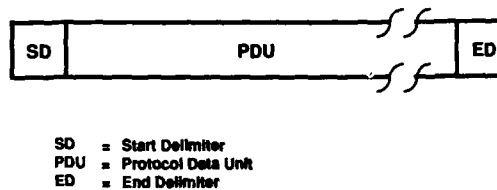
Station failures are handled by the station immediately preceding the failed station in the logical ring sequence. The station passing the token verifies that there is bus activity. After two consecutive attempts at passing the token, the station automatically increments the destination address in the token and tries again. This incremental bridging continues until the station finds a successor or the destination address wraps around and matches the local station address at which time the station shall cease its attempts to find a successor. Stations are allowed admittance to the logical ring on a periodic basis. Each station has a ring admittance timer (RAT). When this timer expires and there is a gap between the local station's address and that of its successor, the token is passed to the sequential address following that of the local station. The normal token passing rules are then applied. Therefore, if any of the stations in the gap desire admittance they shall be granted an opportunity during this time.

Initialization of the logical ring occurs after the token is lost or on power up. Each station which powers up and completes its own internal diagnostic and startup procedures activates a bus activity time (BAT). If the station hears any valid transmission it resets this timer. This indicates that some portion of the logical ring is active and the station defers any activity. The station may receive a valid token addressed to it because the ring admittance timer of a station numerically preceding it in the logical ring has expired. In this case, the station begins to hunt for a successor using the normal token-passing rules. If the bus activity timer expires, the station attempts to gain control of the token. It transmits a frame whose length is determined by the station address and the physical length of the bus. After completing its transmission and waiting for a time based upon bus length it listens. If the station hears any other transmission, it has lost the claiming process and waits for the token. If the station hears nothing it assumes that it has won the token and begins to hunt for a successor. In this case the normal token-passing rules apply.

System monitoring is provided by watching the passage of the token throughout the logical ring and station management status messages. The user is notified whenever a change has occurred or when commanded to do so. Station management status messages are utilized by a station to report specific, non-fatal problems such as a receiver or transmitter failure.

Because the selection of physical medium dependent (PMD) components strongly depends on the characteristics of the physical layer protocol, a description of the line states and symbol set used is needed. There are 2 line states, quiet (or idle) and busy, and 3 kinds of symbols, control, data, and violation. When there is no bus activity, the transmission medium is said to be quiet. It is busy if a signal is being transmitted on the physical medium. A signal is defined by a range in the time rate at which transitions must occur according to the encoding scheme chosen. For example, a 50 Mbps/100 Mbaud Manchester-encoded implementation (the method for the Pave Pillar HSDB and many others) must have transitions every 10, 15, or 20 nanoseconds (ns). (This includes allowed illegal Manchester symbols.) So, there must always be at least 3 signal transitions in any 4 consecutive data bit slots to be considered busy. This is important because the receiver and clock recovery unit may be sensitive to the transition density of the signal.

The structure of every frame transmitted must conform to that shown in Figure 4. Frames may be concatenated up to the limit established. In addition, every transmission is preceded by a preamble used for receiver synchronization. This is the first of the control symbols. The length of the preamble, which depends on a myriad of factors, is given in the "slash sheet" to the standard or in a specification for an intended implementation. A 16 bit-time preamble is sufficient for any Manchester-encoded HSDB in a passive star-coupled topology. It is always a maximum transition density code; thus for the Pave Pillar HSDB, there is a transition every 10 ns up to 320 ns (16 bit times). The start and end delimiters (SD,ED) are the other control symbols. They are unique and define the beginning and end of the protocol data unit (PDU). In Manchester-encoded implementations, the SD and ED must be "illegal" to be unique; that is transitions occur 15 ns apart instead of every 10 or 20 ns for "legal" Manchester code in a 50 Mbps HSDB.



15-488303

Figure 4. Overall Frame Structure of a Transmission.

The PDU carries the data symbols each conveying one or more binary digits (bits). In Manchester-encoded systems, only legal (valid) Manchester symbols are allowed. All "ones" or "zeros" of data give the highest symbol transition density, the same as the preamble. Alternating "ones" and "zeros" give the lowest symbol transition density, one each 20 ns in a 50 Mbps HSDB. Any detected activity which does not conform to prescribed control or data symbols is a violation symbol. The resulting bit errors are grounds for rejection of a frame.

One other parameter of importance to the design of the fiber optic components is the length of time between consecutive transmissions by different transmitters. It must be such that a system minimum intertransmission gap time (IGT) is guaranteed at any given receiver's input. The optical receiver must perform as specified (usually by bit error rate on data bits) when the minimum IGT occurs under worst case operating conditions where the full intertransmission dynamic range (IDR) exists between the end bit of one message and the start bit (preamble) of the next message. With this information, the fiber optic avionics bus interface and interconnect components can be specified and a complete HSDB interface designed. Before we do that, though, a brief look at the rest of the ABI components will be taken.

4. THE PARALLEL INTERNAL (BACKPLANE) BUS PROTOCOL

The avionics bus interface module is one of many housed in an integrated rack serving a variety of users or hosts. They are all connected by a parallel internal (PI) bus which resides in the backplane of the rack. The HSDB is used to tie together various racks which make up the avionics suite. Figure 5 shows a typical configuration including the PI-bus interfaces and the basic internal structure of the ABI. Here is a brief description of its origins and how it works.

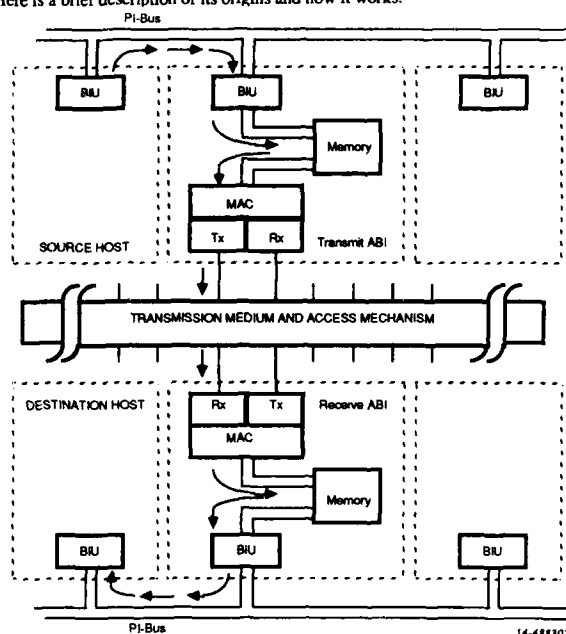


Figure 5. HSDB Network Diagram.

The parallel interface bus evolved from the work of three VHSIC contractors and the U.S. Air Force [10]. The PI-bus consists of VHSIC devices connected at the backplane through transceivers to form a 12.5 MHz bus with a 16 (or 32) bit path width [11]. Unlike the token-passing HSDB, it is a contention bus in which each user or host connected including the ABI must vie for the right to acquire temporary bus mastership. Once a host has obtained bus mastership it can send data to, or receive data from, another host. To accomplish this the bus master sends header information over the bus followed by data. The source host commands the PI-bus "Bus Interface Unit" (BIU) (Figure 2) through a Communication Control Block (CCB) which is accessed through a 1750A XIO sequence or equivalent. This sequence (Figure 6) provides the BIU with two CCB control words. The first (Control Word 1) contains the logical PI-bus priority and the upper eight bits of the address of the CCB chain. The second (Control Word 2), provides the lower 16 bits of the CCB chain. Upon the reception of CCB Control Word 2 the BIU begins processing the CCB chain. This chain consists of control data, the module's data buffer address, and the four PI-bus header words (Figure 7). The first of these header words designated "Header Word A" contains a PI-bus physical address, or the PI-bus broadcast address. The second header word (B) contains a value indicating the number of words to be transferred. The third header word (C0) contains a label, and the fourth (C1) contains an offset.

Header words C0 (HWC0) and C1 (HWC1) are used to address the ABI's memory. In the first transmission to a ABI the label (HWC0) represents the ABI's starting address and HWC1 contains all zeros. If for some reason the message is suspended, the source host must send a resume-sequence header prior to sending the rest of the message. The resume-sequence HWC0 will contain the same label and the value of HWC1 will represent the number of words successfully transferred before suspension. Upon receipt of an unsolicited PI-bus message the ABI's BIU will test the label and, if the label is contained within its label table and is active, accept the message. The ABI will then store the data in its interface memory using the value obtained from the label table for starting address (Figure 8). When the entire message has been received, the BIU, if requested, will notify the ABI's 1750A processor via a "Program Control Interrupt" (PCI). Upon receipt of the PCI the ABI will begin processing the message and will transmit the message across the HSDB subject to instructions received from the host device and the HSDB protocol.

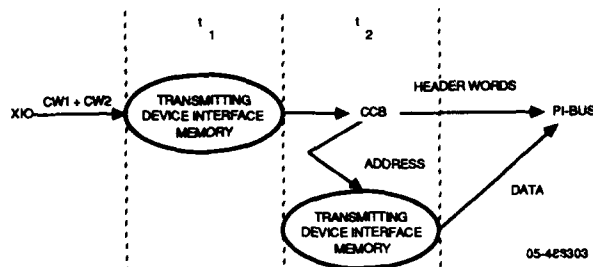


Figure 6. Control Words Initiate PI-Bus Operation.

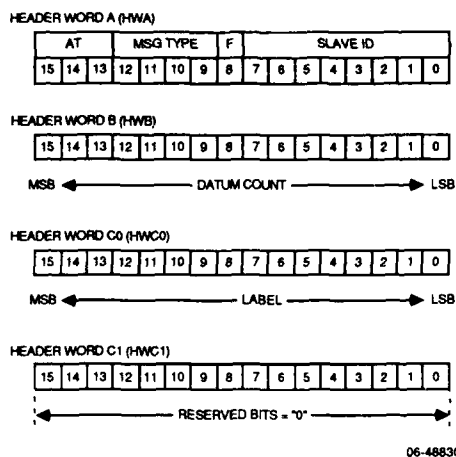


Figure 7. PI-Bus Header Words.

Recall that the PI-bus addressing scheme uses an address (or ID) and a label to communicate across the PI-bus. The address is an ABI's physical, logical, or broadcast address. The user can thus address a specific physical ABI or can use logical addressing (which permits various addressing techniques such as group addressing). However, the user can take advantage of the broadcast address and labels to develop an architecture based upon the powerful technique of global addressing. In this case each process and/or function in the avionics suite has a unique address. Any process or function needing to communicate with another process or function need not be concerned with the physical address or location of the destination since all communication would be through the label. This has significant advantages during reconfiguration due to changing mission requirements, or possibly if there is an equipment failure in that it is only necessary to change a single label at a single location to accomplish reconfiguration. In contrast, if physical addressing were used for a case in which a source process was transmitting to many other processes, changing and verifying all of the destination addresses could be a formidable task increasing the probability of error.

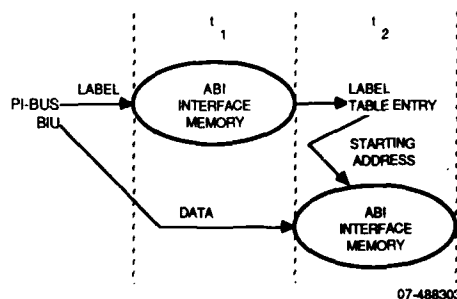


Figure 8. PI-Bus Label Derives Starting Address.

5. THE FIBER OPTIC HIGH SPEED DATA BUS COMPONENTS

In the discussion which follows, we are concerned chiefly with the fiber optic transmitter and receiver circuits of the ABI, shown in Figures 2 and 5. The design of these components is determined by characteristics of the HSDB protocol machine, requirements of the platform in which the HSDB resides (such as an aircraft), and component and configuration options for the HSDB network external to the ABIs. A basic requirement of the Pave Pillar HSDB is to serve up to 64 ABIs. In a high performance aircraft which operates in an environment with many variables, the best design approach is not obvious.

A convenient point of departure is to trade-off configuration or topology options and active versus passive implementations. Active here implies the use of devices to amplify or regenerate a signal along the path between transmitter and receiver. Passive topologies have no "gain" elements between source and sink. In general, the complexity of the added active components can be traded against the complexity of the terminal transmitters and receivers, and the risk, reliability, survivability, maintainability, etc. of the two approaches. Many studies have shown that a passive star-coupled bus has distinct advantages over an active implementation in these respects. Despite the challenge a passive topology presents, the recent development and availability of several key components, and the wise choice of certain design options, makes this approach feasible [12]. In this paper, both passive and active implementations will be considered. We will first consider passive star-coupled networks.

The basic passive star-coupled fiber optic high speed data bus has a single access coupler, the so-called star coupler, at the center of the system. A schematic representation of this network is shown in Figure 9. It generally has the same number of inputs and outputs. A signal coming in on any input is distributed by the device to each and every output in approximately equal amounts. If there are signals on two or more inputs at any one time, they will be mixed optically and then distributed among all the outputs. Thus, to keep multiple signals (at the same optical wavelength) from interfering, there must be transmissions from only one optical source (ABI) at any given time. Since all receivers hear all transmissions, this topology is sometimes referred to as a broadcast network. There are several viable passive access coupler fabrication technologies, but the one which is the most popular at this time and has undergone the most successful environmental testing is known as the fused biconical-taper coupler [13]. This discussion is based on data for a star coupler of that kind. The remainder of the network consists of the ABI transmitters, up to 64, connected to the access coupler inputs with fiber optic cable and optical connectors as required; and up to 64 ABI receivers connected in the same manner to the coupler's outputs.

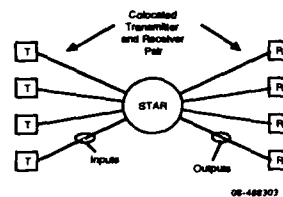


Figure 9. A Star-Coupled Network.

A link power budget analysis approach to network design is usually used. Beginning with the transmitter, the optical power coupled into a fiber is tracked through all the components in the network to the receiver input. After calculating all the system factors, the receiver performance requirements are specified and a separate investigation of the optimum approach to meeting these requirements is necessary. It is not until this point that the multitude of receiver design options, encoding and decoding techniques, preamble requirements, clock recovery issues, intertransmission gap times, and so on, begin to enter the picture. This activity is a major portion of the whole design effort, yet is quite sensitive to the interconnect component requirements and assumptions.

To insure an optimum system design, factors such as achievable transmitter power, thermal effects and compensation, and end of life criteria; connector qualities and kinds, their and cable intrinsic and induced loss and distortion contributions and the fiber size dependence; and access coupler splitting and excess loss, port-to-port variations, and environmental performance expectations must all be taken into account. The weak, distorted optical signal at the ABI receiver input must be reconstructed into a reproduction of the transmitted electrical signal, a clock waveform must be extracted and synchronized to the data, and both presented to the next layer of activity, all done with the fewest number of errors possible since error correction is a process not available in this bus, only error detection (the cyclic redundancy check).

6. THE TRANSMITTER OPTICAL SOURCE

The difference between available transmitter power and receiver sensitivity is the link power budget. The difference between the actual power delivered to the receiver input and the receiver's sensitivity is the link margin. Clearly, to obtain the largest possible link margin, one should choose the largest possible transmitter power. For fiber optic systems in general, that is achieved with the use of injection laser diode (ILD) devices; however their use is not recommended here, not so much because they require large dc power for operation, but because they require more time to turn on from a totally off (unbiased) state than is available in this system. The broadcast nature of this network does not permit unmodulated carrier transmission by multiple sources in the background. So it is necessary to employ light-emitting diodes (LEDs) for this system.

Next, consider the LED-to-optical fiber interface. Since the source size is small, and surface-emitters (SLEDs) have a nearly Lambertian (cosine law) radiation pattern, any effort to concentrate all the energy on the end of an optical fiber will have limited success. (Edge-emitters have some of the same problems as ILDs and are not as attractive.) A simple way of collecting the maximum amount is to use the largest diameter fiber that is practical. More will be said about fibers later, but fiber core diameters up to 200 micrometers are worth considering. Rugged avionics cables incorporating fibers up to this size have been fabricated and are currently flying with success on commercial and military aircraft in this country and abroad. The possibility for even larger fibers in the future exists.

The task of selecting a prototype LED is particularly easy because high radiance devices in hermetic packages with 100 and 200 micrometer core optical fibers pigtailed to the device are currently available. They have been used in many programs and proven to have the reliability needed for this application. One supplier provides devices with room temperature (25°C) peak coupled power of -1.25 dBm and +2 dBm, minimum, for 100 and 200 micrometer core fiber, respectively. Typical power levels are 1 dB greater [14]. The rise and fall times of these devices support a 1000Mbaud system. The final available transmitter power will be less than these figures, of course, because they must be derated to account for a variety of factors discussed next.

The most significant factor is operating temperature. The modules housing the transmitter will be actively cooled, but a junction temperature of 105°C can be expected sometime during the device life. In a very cold ambient environment, considerably lower junction temperatures will exist. Since the LED's power output varies with temperature, it is essential that the output be actively compensated. This requires the maximum output power to be reduced by about 3 dB to accommodate the 80°C rise above room temperature which may be encountered. This preserves the reliability desired for the device. Next, it must be recognized that the output power will decrease throughout the life of the device. An accepted end-of-life criterion for power is 3 dB below the initial level which could occur as soon as 20 or 30 years after installation assuming continuous operation. So far, 6 dB of derating has been applied. There will be a spread in optical power among new devices which will be assumed here to be 3 dB. This value is wide enough that there should not be any significant cost penalty. Noting that a near end-of-life transmitter and a new one could both be in the system at the same time, there could be a swing of 6 dB in acceptable output power from temperature compensated transmitters. (This would increase to perhaps 10 dB if uncompensated.)

Table 1 summarizes the above discussion for the prototype LEDs with 100 and 200 micrometer optical fibers. In both cases, it can be shown that the passive star-coupled high speed data bus is viable with the correct receiver design. Because of the increased available power, and other reasons to be mentioned, it is preferable to use a 200 micrometer core fiber from the optical source. However, until it is necessary to know how much power is incident on the receiver, it is not necessary to make the choice.

Table 1. Available Transmitter Power.

| Optical Power, dBm | Fiber Core Size (μm) | | | |
|--------------------------|----------------------|-------|------|------|
| | 100 | | 200 | |
| | Min. | Max. | Min. | Max. |
| At Room Temperature | -1.25 | +2.25 | +2 | +5 |
| After Temp. Compensation | -4.25 | -1.25 | -1 | +2 |
| At end-of-life | -7.25 | -4.25 | -4 | -1 |

09-448303

7. THE OPTICAL CONNECTORS

Defining the characteristics of the optical connections is one of the most difficult parts of the data bus design because the physical attributes of the intended installation are almost always poorly defined. The problem is significant for fiber optics because, unlike conventional wire, the system performance is sensitive to the kinds of connections and the number of them. Fortunately, several physical characteristics of the avionics packaging plan are well enough defined that some of the uncertainty about the interconnection issue can be removed. The nature of the application suggests that most of the connections will have to be demountable to support maintainability and availability requirements. Quasi-permanent or totally permanent splices are reserved for a few specific places. The use of fusion splices (fusion by electric arc) may be limited to harness fabrication in the wire shop or on the assembly line due to safety considerations. Other splicing techniques will have to offer substantial advantages over demountable connections to justify their use. In the present packaging scheme for future avionics, there will have to be two serial disconnects associated with each transmitter and receiver: one on the ABI module housing them and one at the "rack" housing a number of the ABI modules. This is, at least, the cleanest configuration. There will also be several "bulkhead" disconnects at structural supports or airframe splices. An allowance for two of them will be made here although more are possible if necessary. Finally, there are the input and output connectors at the access coupler. Figure 10 illustrates an installation concept based on the above discussion.

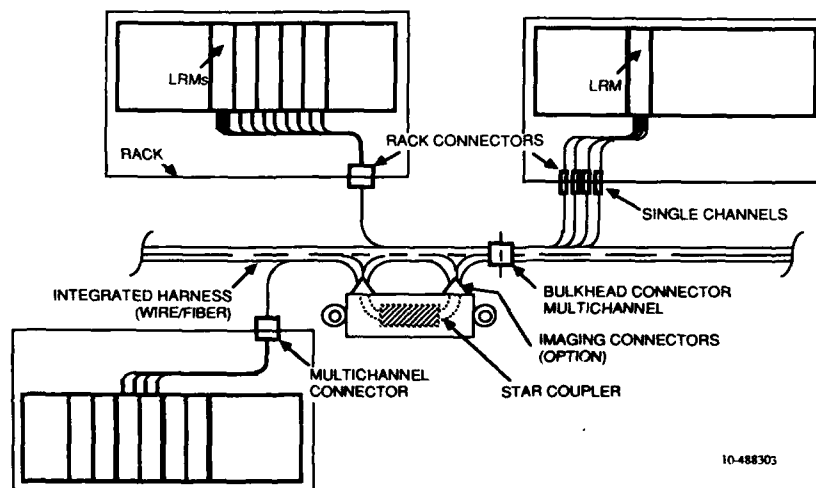


Figure 10. Typical Aircraft Installation.

Assigning a reasonable value for insertion loss for these connectors is a difficult task. It is important to consider both the intrinsic connector loss and the effect that time has on performance. One worst case scenario is to pick some nominal value for insertion loss, add several decibels to each for degradation, and then multiply that by the number of connectors in the path from a transmitter to receiver. With recent improvements in connector technology resulting in lower intrinsic losses, consistency in loss

from mating to mating, and improved alignment mechanisms, it simply is not necessary to take an extreme view of the connector loss situation. Single channel connectors with losses consistently below 1 dB for hundreds of matings and repeatability in the 0.1 dB range are now available from multiple suppliers. A worst case in which all connectors have high loss simultaneously is probably not a reasonable one, especially when consistency in performance is taken into account. Furthermore, after a maintenance action, a good optical connection can be verified with the system's built-in-test circuitry. Once the connection is made, dirt and other foreign objects will be virtually unable to enter the connector space in any reasonable design. Finally, there are connector terminii designs now being fielded which use lenses and still retain small diameters. They have the advantage of being much less sensitive to small obstructions in the space between the connector halves because of their relatively large optical cross-sectional area. This property offers an advantage over connector terminii in which the ends of the fibers face each other directly.

So, the following connector loss assignments are made. For six connectors at the modules, racks, and bulkheads, a total worst case loss of 6 dB is assessed. Thus there might be four 0.5 dB connectors, a 1 dB connector, and one 3 dB connector which has suffered a difficulty, but not one requiring maintenance action. Any other assignment can be hypothesized, of course. For the two connectors at the star coupler, a total worst case loss of 6 dB is assessed. This is done because a different type of connector may be required here to accommodate a large number of connections (64). A greater degree of freedom in losses is a prudent decision for this connector. Collectively, a worst case loss of 12 dB is assigned to the eight connectors allowed between any transmitter and receiver in this network. The lowest loss (best case) will be found for those paths which do not have bulkhead connectors, and which have optimum connections. For analysis purposes, an assignment of 4 dB is made for this case.

8. THE OPTICAL FIBER AND CABLE

The fiber optic cable loss is divided into intrinsic and externally induced contributions. Scattering and absorption are responsible for most of the intrinsic loss which is generally very low (a few dB per kilometer) for modern optical fibers. The cabling and jacketing process is responsible for increasing microbending loss, but can be controlled through careful design. A larger diameter fiber of given numerical aperture resists microbending better adding support to the argument favoring its use. Also, to keep microbending induced problems to a minimum, the ratio of core to cladding diameter (the aspect ratio) should not be too great unless high numerical aperture (NA) fibers are used [15,16]. Other factors discourage the use of high NA fibers; thus an aspect ratio of 0.7 or less is preferable. This limit corresponds to 100/140 or 200/280 core/clad ratio fibers. Despite this rule of thumb, there are higher aspect ratio fibers being used with success on aircraft at this time.

Externally induced loss comes from the effects of bends put into the cable during installation (macro bends). To keep this to a small percentage of the intrinsic loss, it is only necessary to conduct the aircraft cable fabrication and installation process in a manner consistent with the requirements of the technology. Minimum bend radii are dependent on the jacketing construction, fiber diameter, and installation handling including the tensile loading involved in the installation. For nearly any case, minimum values are comparable to those specified for conventional wire of similar cable size, typically an inch or so. Another source of induced loss is the effect of nuclear radiation products, both electromagnetic and ionizing. For avionics applications, both pure and doped silica core fibers have demonstrated their ability to perform in the transient environment, at low temperature, and at the short wavelengths which high power LEDs produce. Fibers with the best performance in this environment have relatively low numerical apertures (0.2 to 0.25). Therefore, if nuclear radiation hardness is required, a fiber must be chosen which is compatible with the other effects discussed.

The optical fiber not only attenuates the signal but also distorts it. Two effects contribute to pulse distortion in multimode fibers: modal dispersion and the material component of chromatic dispersion. These terms do not directly affect the static loss budget or link margin but have a dynamic effect when the receiver performance is included. Thus it is necessary to carefully consider these factors to be sure the fiber which is selected based on other reasons does not create a problem for the system when it is all put together. Modal dispersion results from different modes (rays) taking different paths inside the fiber. In principle, a larger diameter fiber will have a larger modal dispersion, for a given numerical aperture, because optical path differences are greater. This can be offset with lower NAs in larger fibers. Also step-index fiber has greater modal dispersion than graded-index fiber. At 850 nanometers, a typical value for 100 meters of 200 micrometer core semi-graded-index fiber is 2 nanoseconds. Care must be exercised in interpreting manufacturer's "fiber bandwidth" specifications since the actual dispersion which results is sensitive to both the length (it is not linear) and the test or launch conditions for the fiber.

Material dispersion results because different wavelengths (colors) travel through the fiber at different speeds. Thus the broader the spectrum of an optical source, the greater the dispersion. This factor also decreases with increasing wavelength to a point. Since 850 nanometer LED sources, with spectral widths in the order of 50 nanometers, are preferred for this system because of the large available transmitter power, the material dispersion component is not negligible. At this wavelength, a typical value is 0.1 nanoseconds per nanometer per kilometer. For a 50 nanometer spectral width and 100 meters of fiber, the material dispersion is 0.5 nanosecond. The net dispersion is the root-sum-squared value of the two components or 2.06 nanoseconds. Summing this in a like manner with the transmitter's rise and fall times results in a value for the rise and fall time of the optical pulse incident on the receiver's photodetector. This data is then used to determine the receiver bandwidth required to insure an "eye" opening adequate to maintain a specified probability of detection (bit error rate).

For aircraft applications, the cost of 200 micrometer core optical fiber should not be significantly greater, if at all, than the smaller 100 micrometer fiber since the raw material cost is not dominant, and the process for manufacturing larger fiber is not much different. With both sizes of jacketed fiber optic cables now flying on commercial and military aircraft in different parts of the world, there should be little reluctance to embrace either as a standard. With the advantage the larger fiber offers in terms of modal volume and coupled power, it is the preferred choice. Typical examples of fiber optic cable losses are zero decibels for transmitter-to-receiver path lengths near zero and 2 dB for the longest path which has the highest intrinsic and macrobending loss and a 1 dB penalty for radiation induced degradation. These are reasonable values for path lengths that will not exceed 100 meters.

9. THE ACCESS COUPLER

Fused biconical-taper star couplers with up to 100 inputs and outputs were being fabricated [17] shortly after the announcement of the concept [13]. They can be made with any number of inputs and outputs, but standard products usually have a power-of-two of them (2, 4, 8, 16, etc.) [18]. Larger couplers are not necessarily more difficult to make than small ones; they just have higher splitting loss - 3 dB more for every doubling of the number of inputs/outputs. The larger number of fibers also means more connections which impacts the physical requirements for housing the coupler. Most couplers manufactured to date are made with 100 micrometer core fiber or smaller. However, couplers with 200 micrometer fiber have been made and may be producible with higher yield. The larger fiber size also means the uniformity of output power will be better and the overall excess loss generally will be less if the right fiber is used.

Recent tests of commercial-grade access couplers to Class 3 avionics equipment environmental exposures have been particularly encouraging [19]. The tests included mechanical shock, vibration, thermal shock, thermal range, humidity, and salt spray. The thermal shock test ranged from -55°C to +125°C. The coupling factor, backscatter, excess loss, and output uniformity were observed. Despite these units being intended for commercial applications, several passed all Class 3 tests and showed little degradation as a result. The 32 x 32 port coupler which passed also had the best performance prior to testing. Clearly, these components of the system are suitable for the avionics environment and can be militarized without extensive investment. The best 32 x 32 port coupler in the tests had a mean coupling factor of 16.7 dB, just 1.7 dB above "ideal" (the ideal being $10 \log 32 = 15$ dB). Output uniformity (the difference in dB between the maximum and minimum coupling factors) was only 1.1 dB. After passing all tests, the mean coupling factor was 17.7 dB, and uniformity 1.6 dB.

A high quality 64 port coupler fabricated with 100 or 200 micrometer core fiber will have superior uniformity figures and be less likely to have a "hot" fiber. Poorly fabricated couplers have considerably more power in the output fiber, corresponding to the input being driven (the hot one), than in the other output fibers. The following coupler loss values are representative. The ideal loss is $10 \log 64 = 18$ dB. For the worst case coupler, it is assumed there is a "hot" fiber and the coupling factor or loss for that path is 17 dB. Thus, a particular transmitter is connected to one receiver by a path that has only 17 dB loss through the coupler. The port-to-port uniformity, excluding the hot fiber, is 2.5 dB for the 200 micrometer fiber coupler, a value smaller by about 0.5 dB from that for a coupler made with 100 micrometer core fiber. The total range of coupling factors or loss is therefore a minimum of 17 dB and a maximum of 20.5 dB. This concludes the consideration of all components except the receiver. Before any particular receiver design is chosen, however, several system parameters need to be considered. Numerical values for them are derived from the foregoing discussion.

10. THE INTERTRANSMISSION DYNAMIC RANGE AND RECEIVER OPERATING RANGE

Before choosing a receiver, it is necessary to understand the dynamic environment in which it must work. Some designs are simply not suitable to data buses, and others are questionable. So, as many variables as possible must be understood before committing to a technical approach.

Without introducing any new numerical quantities, the intertransmission dynamic range (IDR), and required receiver operating range (ROR) can be determined. This bus operates by consecutive transmitters sending bursts of information of varying length. A minimum length of time between consecutive receptions is specified to insure the quality of the transmissions. Thus any given receiver gets these receptions via different paths with different losses. The maximum range of peak optical power which can occur for this situation in a particular configuration is called the IDR. In addition, if a single receiver design is to be used at all terminals it must handle an even greater dynamic range to accommodate all transmissions (not just consecutive ones) over all operating conditions, from the best case to the worst case. This range is called the required ROR. The two quantities will now be derived based on the data previously presented.

The IDR is the difference in loss for the paths to any receiver with the highest and lowest losses. To find the numerical value for the system in this study, it is only necessary to add up the differences for each of the components involved. But what components are involved? Since the path to any given receiver from any two transmitters is the same from the output of the access coupler to the receiver, the components in this path do not contribute to a difference in loss. So what must be found is the lowest and highest loss paths from two transmitters to and including the access coupler. The low loss path in this design example includes the highest power transmitter, the fewest connectors (three: at the transmitter, rack and coupler), the lowest connector losses, essentially zero fiber length, and the lowest coupler loss. The high loss path has the lowest power transmitter, the most connectors (four: at the transmitter, rack, one bulkhead, and coupler), the highest connector losses, the maximum fiber length (100 meters), and the highest coupler loss. Table 2 lists the numerical values associated with each path. Note that there are no absolute power levels (dBm); only differences (dB) are needed to find the IDR. A 6 dB difference in LED power was previously computed for the worst case (temperature compensation is assumed). For the connectors other than at the coupler, a 2.5 dB difference is assigned, and at the coupler's input connector 1.5 dB. The fiber loss ranges from zero to 2 dB, and the access coupler loss varies from 17 to 20.5 dB. The use of 200 micrometer core fiber is assumed in this example. The result is a 15.5 dB IDR.

Table 2. Calculating the Intertransmission Dynamic Range (IDR).

| | Loss Contribution (dB) | |
|------------------|------------------------|-------|
| | Best | Worst |
| Transmitter | 0 | 6 |
| Connectors | 2 | 6 |
| Fiber | 0 | 2 |
| Access Coupler | 17 | 20.5 |
| TOTAL | 19 | 34.5 |
| Difference (IDR) | 15.5 | |

11-488303

The required receiver operating range is calculated in a similar manner, but now the lowest and highest loss paths from the access coupler outputs to receivers must be included. By adding the value of the lowest transmitter power to the worst case path loss, the minimum power incident on the receiver is known. Any link margin added to that specifies the sensitivity required for a single receiver design to support the entire data bus. The development of these quantities is illustrated in Figure 11. From Table 1, a 6 dB range of temperature compensated transmitter peak powers from +2 dBm to -4 dBm may be found. The range of losses for the connectors other than those at the access coupler is 1 dB to 6 dB. Note that there may be no bulkhead connectors in a given path; thus the loss for them is zero decibels. The coupler's two connectors range from 3 dB to 6 dB. Together, all connectors are allowed a loss range from 4 dB to 12 dB. The fiber best case loss is zero dB for a short path; the worst is 2 dB for a 100 meter path with a permanent loss contribution of 1 dB due to radiation effects. The access coupler loss is the same here as above: 17 dB to 20.5 dB. The results are summarized in Table 3. The receiver will have to have a sensitivity equal to or greater than the worst case value. Its operating range is 19.5 dB and extends from -38.5 dBm to -19 dBm. With this information, the process of matching the interconnect system to a receiver can begin.

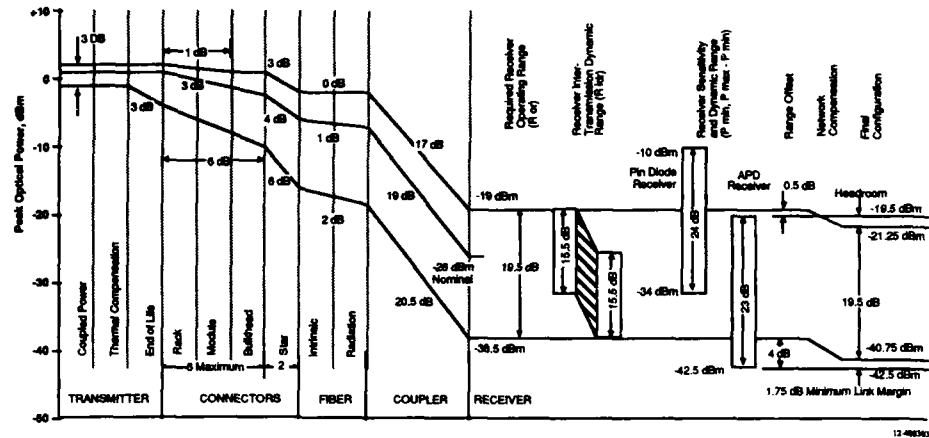


Figure 11. Development of the Power Budget and Matching it to the Optical Receiver.

11. THE HSDB RECEIVER CHOICES

The results so far indicate that a high sensitivity, wide dynamic range receiver will be required to support this system. As little as -38.5 dBm peak may be incident, and in the worst case the input could range over 19.5 dB. So what kind of receiver will fulfill the requirement?

Fiber optic receivers for the HSDB can be classified as ac-coupled, edge-coupled, or dc-coupled. Refer to Figure 12. All 3 typically incorporate a photodetector and band-limited transimpedance preamplifier in the "front-end." The distinction between types begins at this point. The first two designs use ac-coupling prior to data detection. The first is generally referred to as an ac-coupled receiver. The front end is followed by additional gain stages as required. The output is capacitively coupled to a zero-reference comparator or hard limiter for waveshaping and amplitude matching to a standard digital interface such as emitter-coupled logic (ECL). A long time constant is used in the capacitive coupling to preserve the optical waveform to some defined extent. The dc offset buildup in the gain stages is removed in the ac-coupling.

Table 3. Calculating the Required Receiver Operating Range (ROR) and Minimum Input Power.

| | Best | Worst |
|--------------------------|---------|-------|
| Transmitter Output (dBm) | +2 | -4 |
| Connector Loss (dB) | 4 | 12 |
| Fiber Loss (dB) | 0 | 2 |
| Access Coupler Loss (dB) | 17 | 20.5 |
| Receiver Input (dBm) | -19 | -38.5 |
| Receiver Operating Range | 19.5 dB | |

13-488303

The second ac-coupled type is called an edge-coupled or differentiating receiver. The front end output is ac-coupled with a short time constant to a post amplifier and Schmitt trigger with hysteresis. The output is buffered as required to provide a standard digital logic interface such as ECL. The preamplifier's dc offset is removed by the differentiating capacitors. The post amplifier's dc offset is removed by the hysteresis in the data threshold detector (the Schmitt trigger). Hysteresis is required to prevent false triggering on noise.

A fully dc-coupled receiver has no capacitive coupling between the optical input and data detector. Since there is a large amount of dc offset at the input to the data detector, some form of edge-detecting process is required. A threshold must be established based on the average value of the input waveform in order to make bit decisions. Usually, several delay lines, gain stages, logic elements, and comparators are necessary to implement this circuit. It is often called an edge-detecting receiver (not to be confused with edge-coupled receiver).

A few general comments on the receivers follow. For optimum performance, the frequency response of the preamplifier must be matched to the rest of the network. Specifically, since the transmitter output is band-limited, the receiver must be compensated correctly. Thus the transmitter and receiver designs must be considered together. Also, the frequency response of the preamplifier may affect the overall receiver performance depending on the circuit which follows the low-pass filter. Thus the same transimpedance preamplifier may not be the best choice for all three receiver types. In general, a specific receiver design is driven primarily by the requirement for a quantitative measure of performance given by the probability of error (bit error rate). That in turn is qualitatively indicated by the "eye" pattern the receiver output produces. The "openness" of the eye is determined by numerous factors only a few of which have been indicated here.

The choice of receiver type is driven by several factors including certain properties of the protocol discussed in Section 3. To achieve the high sensitivity and wide dynamic range operation for the passive star-coupled data bus, the ac-coupled receiver is the most attractive. It has limitations, but they are not in sensitivity and dynamic range. This type only works well with signals obeying constrained run-length (CRL) codes and optimum performance is obtained when the duty cycle is exactly 50%. A sensitivity penalty is paid if the steady-state duty cycle is different. However, CRL codes such as the 4B5B + NRZI or 8B10B + NRZI block codes, which have short term duty cycle variations from 40 to 60%, have a long term duty cycle which approaches 50% as the message

length increases. The ac-coupled receiver has little difficulty with these data codes and the sensitivity penalty resulting from the slight amount of "baseline wander" which results is small, less than 1 dB, typically.

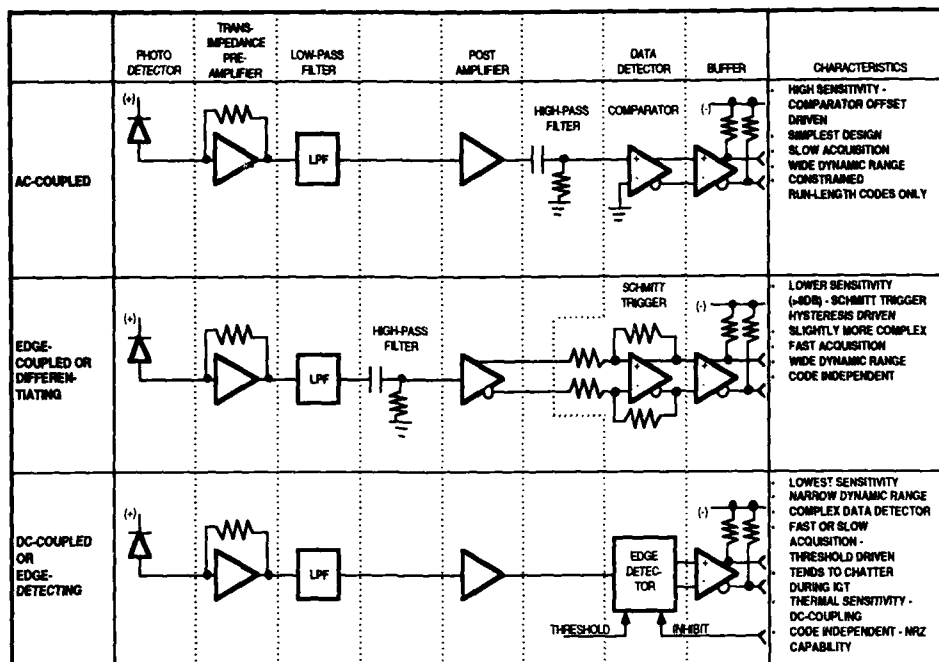


Figure 12. The HSDB Receiver Choices.

16-488303

Since the receiver must work in a burst mode environment over a potentially large (15.5 dB) intertransmission dynamic range, it must be given sufficient time to adapt to each new signal level. The receiver thus has an acquisition time associated with it after which it responds properly to the CRL code for which it was designed. The acquisition time is a function of the time constant of the high-pass filter and is accommodated by the combined intertransmission gap time and preamble length (in time)[20]. The purpose of the preamble is to get the receiver's operating levels established and the clock recovery circuitry running in phase with the data; so it is a contributor in the acquisition process. Thus an ac-coupled receiver best serves the need of this data bus as long as sufficient acquisition time is provided and an appropriate CRL data code is used.

For maximum sensitivity, the receiver bandwidth must be minimized. The bandwidth required to achieve a given level of performance (bit error rate) is determined by the quality of the optical signal incident on the receiver as measured by its rise and fall time and pulse width distortion. The higher the quality of the input, the lower the bandwidth needed to produce a given BER. The data code used directly drives the bandwidth needed for the entire data channel including that of the receiver. For example, the block codes mentioned above impose a 25% penalty on required signaling rate over the NRZ data rate whereas a code such as Manchester, usually used for low data rate systems, imposes a 100% penalty (i.e., the signaling (baud) rate is twice the NRZ data rate). At the data rates involved in this system (50 Mbps) this translates into a 3 dB sensitivity penalty for Manchester compared to 8B10B + NRZI. Nevertheless, a requirement in the Pave Pillar HSDB and related buses is the use of Manchester encoding, and a receiver with bandwidth sufficient to support a 100 Mbaud signaling rate is necessary.

12. THE RECEIVER DESIGN

The system measure of performance, the bit error rate, is a function of the signal-to-noise ratio (SNR) at the receiver input and the offset and hysteresis performance of the data detector. The principal noise sources are diode shot noise, the preamplifier input transistor thermal and shot noise, and the feedback resistor thermal noise. The diode contributes the least to the total since its noise current is proportional to the square root of the dark current in a pin photodiode, and to a portion of the dark current multiplied by a gain factor for an avalanche photodiode (APD). The preamplifier's noise, being directly related to bandwidth, is minimized by keeping the bandwidth as small as possible. The amount of signal necessary to achieve the SNR corresponding to the desired BER is the preamplifier's sensitivity.

A detailed receiver design analysis is not within the scope of this paper. The reader is referred to any of a large number of articles (such as [21]) which allow a designer to quantify the receiver performance. The results of a HSDB receiver design at the Harris Corporation which was then followed by fabrication and a performance evaluation is presented here and used to complete the receiver matching process. An ac-coupled receiver with pin photodiode, optimized for operation at 100 Mbaud with Manchester-encoded data, designed to operate up to 105°C junction temperature, and packaged to achieve the lowest noise figure, can achieve a -34 dBm (peak) sensitivity at 10^{-10} BER and 24 dB dynamic range for 50% duty cycle signals. Unfortunately it does not have enough sensitivity to meet the system requirement of -38.5 dBm. Refer to Figure 11.

To make this network work with a pin photodiode receiver, it is necessary to reduce link losses or increase transmit power by 4.5 dB. Lower link loss is best achieved by reducing the number of connectors in the system. Designing the platform with fiber

optic technology in mind may make this possible. For example, the use of permanent cabling splices where an airframe is permanently spliced would help. Increased transmitter power is also a possibility in the foreseeable future. However, yet another alternative exists: use an avalanche photodiode receiver. Although an APD requires a low current, high voltage bias supply, the device reliability is no longer questioned. It is important, however, that the APD not be biased for too much avalanche gain. Otherwise its thermal properties will cause the gain to decrease at high input power levels (which is not necessarily bad), but if followed by a low input power signal, the gain will not be sufficient for normal receiver operation. This thermal effect is minimized by biasing the diode for a low gain (about 10). If too low a bias voltage is used, the diode's bandwidth decreases drastically which is not acceptable either. Thus the APD works best in this application when optimally biased. Replacing the pin diode with an APD yields -42.5 dBm sensitivity for 10^{-10} BER and 23 dB dynamic range. This is more sensitivity than required, but, as was pointed out, if it is reduced by reducing APD gain, bandwidth will be sacrificed. The best engineering solution to the range offset is to compensate the link power budget and match it to the best available receiver.

The match can be achieved in several ways. One is to reduce transmitter power across the board. As Figure 11 shows, a link budget downward compensation of 2.25 dB centers the required 19.5 dB receiver operating range in its 23 dB dynamic range and leaves a 1.7 dB headroom and link margin. So the LEDs can be further derated by 2.25 dB. Another option is to use 100 micrometer core optical fiber and increase the expected connector loss by several decibels. Optical attenuators could also be added, but that is not a recommended approach. The link margin which was derived for this system was 1.75 dB minimum. That means that in the worst case with lowest transmitter power, and highest connector, cable, and coupler losses, there will still be 1.75 dB more power incident on the receiver than is required for a specified BER, 10^{-10} here. Noting that the slope of the BER versus signal-to-noise ratio curve is approximately 4 orders or magnitude per dB at 10^{-10} BER, this system, with only -40.75 dBm incident, will have a BER of about 10^{-17} . Under more typical operating conditions, one might expect intertransmission dynamic ranges of 7 to 10 dB, and a nominal receiver input power of -26 dBm peak. In this case the link margin is $42.5 - 26 = 16.5$ dB.

We have shown, based on measured component performance data, that a passive star-coupled serial high speed data bus physical layer suitable for use on future military airborne platforms, and meeting the requirements set forth by the Pave Pillar requirements can be built. Next, we wish to show still another implementation of the physical layer hardware, one based on the use of a repeater or regenerator (an "active" star) instead of a passive access coupler.

13. THE ACTIVE STAR-COUPLED NETWORK

In the most elementary active star-coupled network, the passive access coupler shown in Figure 9 is replaced by a repeater or regenerator station. It has the same number of optical inputs and outputs and also needs power to operate.

Before we consider the contents of the active star, let us examine why we would use it. We already showed how a passive implementation of the HSDB can be built with a nominal 16.5 dB link margin, that is up to 16.5 dB of loss can be added to a normal operating path and the network will still operate as designed, that is it will meet the required bit error rate performance. Possible reasons for using it include: the APD photodetector cannot be used because of the size, weight, and power penalties of its power supply (10 dB penalty, 6.5 dB net link margin); plus the need for large numbers of connectors (4 more at 2 dB loss each, -1.5 dB net link margin); or the use of high loss, non-state of the art connectors (add 8 dB loss to the baseline, -1.5 dB net link margin); or the requirement to operate at a different wavelength (such as 1300 nm) where the increase in receiver sensitivity due to higher responsivity pin photodiodes cannot offset the considerably lower coupled optical power available from LEDs (10 dB net penalty with a temperature compensated transmitter, -3.5 dB net link margin); and so on. Whatever the reason or reasons may be, it is possible to recover lost link margins by the use of a repeater or regenerator in place of the passive star coupler.

When this is done, note the following properties of the fiber optic network. The ABI receivers are effectively moved into the active star so there will be more incident power on the detector. The receiver operating range will be reduced since there are fewer path options, but the intertransmission dynamic range is essentially unchanged. Actually, the ROR and IDR are equal if the active star can be built with one receiver. The ABI transmitters are effectively moved to the output of the active star so all connections from the active star to all ABI receivers are point-to-point links. This means there is no IDR requirement for the ABI receivers any longer. All transmissions to any one receiver will have the same short term peak power level. However, since the paths to all receivers may have different losses, the ABI receivers must have a non-zero operating range if one type is to be used at all terminals. Thus there is some relaxation of performance requirements for the receivers that suggests a reconsideration of the type for an active network implementation.

14. THE ACTIVE STAR TECHNICAL ISSUES

When designing an active star, the overall goal is to insure that the performance degradation of the HSDB is minimized. This means the net data throughput rate must not be reduced significantly. This could occur if the propagation delay through the active star, which is affected by receiver acquisition time, is large; if there is significant pulse width distortion and jitter forcing the active star to employ clock recovery techniques to retune the transmitted signal; or if undesirable changes to the protocol are necessary to implement the active star. The active star must also have adequate optical signal gain to provide a network link margin meeting the system needs. This is met by a careful balance of other link budget parameters, especially transmitter power levels and receiver sensitivities at all locations (ABIs and active star). Most of the factors mentioned interact in some way, so it is not possible to create an optimum active star design without going through an iterative process.

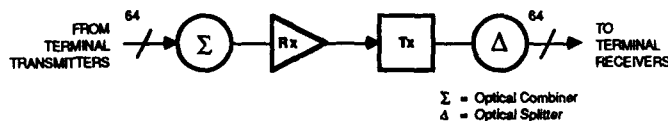
Propagation delay through the active star is minimized by making the receiver acquisition time small. From Figure 12, the clear choice here is the edge-coupled receiver which offers a good compromise between sensitivity and acquisition time performance. However, the 8 dB or more penalty in sensitivity over the ac-coupled receiver can erase any benefit the active star offers in optical signal gain unless an efficient method of active star receiver utilization is employed. Most clock recovery circuits also have an associated acquisition time and will contribute to the propagation delay. The best case is not needing a clock recovery circuit for retiming, but that will be driven by the pulse width distortion and jitter performance of the two concatenated data links (ABI TX to active star RX, and active star TX to ABI RX). Any logic required between the active star receivers and transmitters will add delay. The transmitters themselves only add a few bit times.

Should either an ac-coupled receiver be used, or a clock recovery unit be needed, the non-zero acquisition time may require that either the active star reconstruct the message, or the ABI's transmitted preamble be lengthened. After the active star's receiver acquires the optical signal, the ABI receiver must acquire the active star's output. But the active star receiver "absorbed" a piece of the preamble and the ABI receiver must acquire with a shortened preamble. If reclocking is also required in the repeater, additional preamble will be lost during the clock recovery unit acquisition time. For the single active star configuration shown in Figure 9, two solutions exist. Change the protocol by increasing the preamble length (from 16 to 32 bits, for example) to satisfy the worst case.

This approach results in the smallest hardware impact but does require a minor protocol change which may be unacceptable. The alternative is to decode the incoming message and append a new preamble and start delimiter to it. This method requires more hardware including a 20 bit/40 baud storage register. Message reconstruction eliminates distortion problems, however, and since the hardware penalty will be small if the logic is implemented in a gate array, this approach is reasonable. The major difficulty with it is the demand placed on the clock recovery unit that is needed. The clock signal, derived from incoming data, must now be used to clock data through all the logic hardware to the active star's transmitter. Included in this hardware is a 400 ns long storage register which contains the last 40 baud of the message. Thus, at least 40 clock cycles are required after the end of incoming data. This is a difficult, if not impossible, task for a fast-acquisition clock recovery unit. Alternatively, the storage register could be implemented as a FIFO (First-In, First-Out) memory and a continuous clock (crystal oscillator) could be used on the transmit side. However, a high speed FIFO memory will require more space and additional power which may make this approach even less attractive.

On the subject of pulse width distortion, presently available high-radiance LEDs, with 3-5 ns rise and fall times operated at 10 ns baud times, even with compensation, will have some pulse width distortion of the transmitted waveform. The optical receiver may also add to the overall distortion and jitter. Therefore, the non-ideal output from a transmitting ABI may be further degraded in an active star and be difficult to decode by the ABI receiver. A wide bandwidth receiver and high-speed LED transmitter in the active star will minimize the problem, but wide bandwidth limits receiver sensitivity, and available high-speed LEDs have limited output power. If a message is digitally reconstructed and retimed in the active star, the resulting output is, obviously, as good as that from the ABI transmitter. The need for retiming must be determined.

Active star optical signal gain is determined by (1) optical combining techniques prior to the receiver input, (2) receiver sensitivity, (3) transmitter output power, and (4) transmission splitting loss. Refer to Figure 13. The 64 input fibers must be combined into one optical receiver input for maximum optical gain. An "off the shelf" approach is to use one output of a 64 x 64 passive star coupler. Here, most of the optical power is lost in the 63 unused fiber outputs. One way to increase gain is to use multiple receivers preceded by smaller passive stars. The receiver outputs are logically ORed into a single electrical output. Other approaches include cutting the 64 x 64 passive star coupler in half and coupling the truncated taper region output directly to a photodiode, or coupling a close-packed array of 64 fibers directly to a large area photodiode. If either of these schemes work, the active star will require only one receiver. Note that here the splitting loss of a 64 x 64 passive star coupler will be largely avoided thereby greatly reducing the losses between an ABI transmitter output and the active star receiver.



01-488303

Figure 13. The Active Star Components Which Determine Optical Signal Gain.

On the output side of the active star, the situation is similar. In the simplest scheme, the power output of one transmitter is divided by 64, but the losses are unacceptably high. There are two ways this loss can be reduced: (1) use multiple transmitters with smaller optical splitters, and/or (2) combine the splitter and LED for more efficient optical coupling. The first approach is "off the shelf" but not viable. The second approach requires further development, but significant improvement in efficiency would be achieved. Here, the idea is to exploit the fact that much of the energy from high radiance surface-emitting LEDs is wasted. Ball lenses help to minimize loss but cannot focus all the light into moderate sized optical fiber cores. A larger core fiber or close-packed fiber array will clearly capture more of the LED output. The issue of uniformity among output channels must be considered, obviously.

Recall that there was no intertransmission dynamic range requirement for the ABI receivers in an active star configuration because all connections between the active star transmitters and ABI receivers are point-to-point. That suggests the possibility of using injection laser diodes (ILDs) in the active star. They would have to transmit an idle pattern during the normally "dead" time on the bus. Modifications to the protocol machine in the ABI would thus be necessary, but a considerable increase in coupled output power from each transmitter results. Fewer transmitters are thus needed and savings on size, weight, and power are possible.

15. THE ACTIVE STAR DISCUSSION

Based on experience gained designing and building fiber optic active stars[22], a highly-efficient optical combiner (refer to Figure 13) in conjunction with a large area photodetector is practical. An edge-coupled (differentiating) receiver (refer to Figure 12) provides sufficient sensitivity for a large link margin on the "left" side of the network (refer to Figure 9 or 13). Using the data in Figure 11, the ABI transmitter delivers +1 dBm typical into a 200 micrometer core-size fiber. With half the connector and fiber loss (for the "left" side only), and no coupler loss with a high-efficiency optical combiner, there is -3dBm typical incident on the detector. Using an edge-coupled receiver with -26dBm (peak) sensitivity provides an "artificial" link margin of 23dB. This is not practical because the incident power level is already at the top of the receiver's operating range for typical conditions. Transmitter power must be reduced and higher interconnect losses must be assured. The typical incident power should be near -10 to -15 dBm which provides an 11 to 16 dB link margin (the same as the passive star coupled bus with APD)!

Since the available power from LEDs for the active star transmitters is limited, it is logical to use high sensitivity ac-coupled receivers with pin photodiodes in the ABIs. This means acquisition time is important and a full 16 bit preamble must be delivered to the ABI receiver. If a nominal 16 dB link margin is desired on the "right" side of the network, the incident power on the detector ought to be no greater than -18 dBm. This still provides a 5 - 8 dB margin on the high power side of the receiver operating range. (This assumes a 21-24 dB total ROR.) If link losses are the same on both "sides" of the network (4 dB nominal), then the active star must couple -14 dBm into each of the 64 output ports. With a carefully designed optical splitter (see Figure 13), it is possible to use as few as 4 transmitters with the LED type used in the ABIs. If no clock recovery circuitry is needed, the active star internal components will be so small that the fiber optic connectors totally dominate the size of the overall active star package. Propagation delay is minimal and no modification to the protocol should be necessary.

16. FINAL COMMENTS

The avionics bus interface module provides the gateway between a parallel internal (PI) bus for a "rack" of processors, converters, etc., and the multiplex high-speed fiber optic data bus which ties together all the "racks." The fiber optic bus is thus the backbone for the command and control of the avionics subsystem of an aircraft. An overview of the gateway function has been presented with some detail regarding the origin and nature of the different protocols used for the PI-bus and high speed data bus. Emphasis has been placed on the design process used to implement the fiber optic network. Numerous options are available each chosen based on specific application requirements. Both passive and active star-coupled implementations have been considered in great detail. A modestly-sized subsystem can be implemented with very low risk (adequate link margin, high reliability) using pin photodiode-based ac-coupled receivers. Full-sized networks (64 ABIs) with numerous connectors can still be implemented passively using properly biased avalanche photodiodes in conjunction with ac-coupled receivers. The increase in risk is small being driven exclusively by the reliability associated with the low-current, low-power bias supply for the APD.

The passive versus active star-coupled network trade-off has a particularly interesting result. When the impact of an active star on network data throughput is used as the most important factor -- the one which must not be compromised -- it is difficult to show a significant advantage for an active implementation over a passive one. A detailed reliability study is likely to show a preference for a passive star-coupled network. This conclusion may not be valid, of course, for an architecture, data bus protocol, network configuration, or measure of performance other than the one selected for this discussion. It appears to be valid for the Pave Pillar HSDB and for its most practical configuration, a single star-coupled network.

Although packaging of the ABI has not been discussed, except that it is one of many modules in a rack, this is a most important aspect of the Pave Pillar architecture. The Standard Electronics Module - format E (SEM-E), in varying widths, has been selected to house most avionics functions in the future. A photograph of such a module is shown in Figure 14. Note that all connections pass through a single backplane connector, including power and ground, discretes, and the fiber optic connections to the HSDB. There are now several suppliers developing connectors for this function, but their designs are currently not interchangeable or interchangeable.

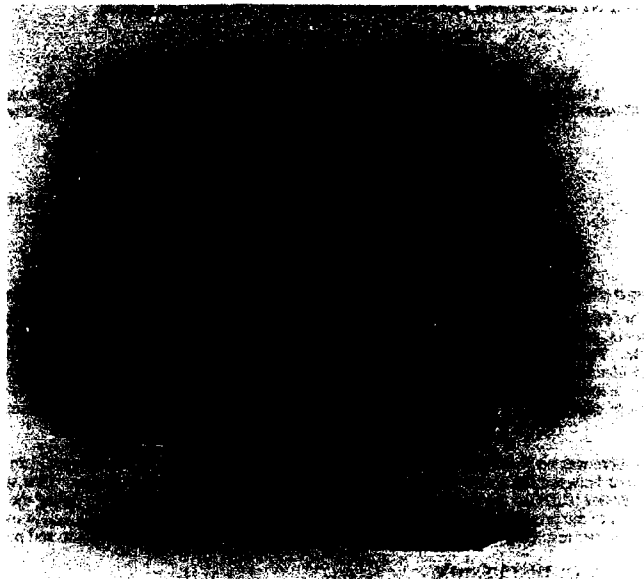


Figure 14. An Avionics Bus Interface Module.

Several manufacturers are currently building ABI and other modules for demonstration and validation of the Pave Pillar avionics architecture. This is the next logical step toward its acceptance and use in the next generation avionics suite. It is our hope that this paper will stimulate additional thought on the subjects presented which in turn leads to an acceptable implementation standard for all.

17. REFERENCES

1. United States Air Force, Air Force Systems Command, "Architecture Specification for the Pave Pillar Avionics," AFWAL-TR-87-114, 1 May 1986.
2. International Organization for Standardization (ISO), "Basic Reference Model for Open Systems Interconnection (OSI)," ISO7498, Geneva, Switzerland, 1983, (available through ISO and ANSI, New York); Zimmermann, H. "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," IEEE Transactions on Communications, Vol COM-28, No. 4, April 1980, pp 425-432.
3. ANSI/IEEE, "Logical Link Control," ANSI/IEEE Standard 802.2 - 1985, ISO/DIS 8802/2, "Token-Passing Bus Access Method and Physical Layer Specifications," ANSI/IEEE Standard 802.4-1985, ISO/DIS 8802/4, Approved 17 December 1984.
4. Society of Automotive Engineers (SAE), Aerospace Systems Division, Committee AS-2, "Linear Token-Passing Multiplex Data Bus," Draft Standard AS4074.1, Version 4.0, 25 January 1988. (Task group member working copy; Current configuration available from SAE, Warrendale, PA., U.S.A.).

5. United States Air Force, Air Force Systems Command, "Architecture Specification for the Pave Pillar Avionics," AFWAL-TR-87-114, 1 May 1986.
6. Lockheed Aeronautical Systems Company, "Avionics Bus Interface Module," Dwg. No. 50R5048, 27 March 1987.
7. Collins Government Avionics Division, Rockwell International, "High Speed Data Bus System Specification," prepared for Pave Pillar program Office, Document No. 10-03, Version 3, Contract No. F33615-83-C-1036, 31 October 1986.
8. Nelson, J.H., et al., "A Candidate for Linear Token-Passing, High-Speed Data Bus Systems," Presented to the SAE Aerospace Systems Division Meeting, Dallas, TX, 4 November 1987.
9. Standardization (NATO) Agreement (STANAG), "High Speed Data Transmission Under STANAG 3838 or Fibre Optic Equivalent Control," STANAG 3910, Rev. 1.3, 27 August 1987.
10. IBM, Honeywell, and TRW, "VHSIC Phase 2 Interoperability Standards: PI-Bus Specification," Version 2.0, 8 September 1986.
11. Snow, J.C., "A User Protocol for a Fiber-Optic High-Speed Serial Data Bus," Proceedings IEEE 1987 National Aerospace and Electronics Conference, NAECON 1987, IEEE 87CH2450-5, Vol. 1, pp. 188-191, May 1987.
12. Uhlhorn, R.W., "Design of a Passive Star-Coupled Fiber Optic High Speed Data Bus for Military Aircraft," Proceeding IEEE 1987 National Aerospace and Electronics Conference, NAECON 1987, IEEE 87CH2450-5, Vol. 1, pp. 158-164, May 1987.
13. Kawasaki, B.S., and K.O. Hill, "Low-loss Access Coupler for Multimode Optical Fiber Distribution Networks," Applied Optics, Vol. 16, No. 7, pp. 1794-1795, July 1977.
14. Plessey Product Data Sheets for type CXL044 and CXL061 Light Emitting Diodes, Plessey Research Ltd., Caswell, England.
15. Olshansky, R., "Propagation in Glass Optical Waveguides," Rev. Mod. Phys., Vol. 51, pp. 341-367, April 1979.
16. Gloge, D., "The Optical Fibre as a Transmission Medium," Rpts. Prog. Phys., Vol. 42, pp. 1777-1824, November 1979.
17. Rawson, E.G., and Bailey, M.D., "Bi-taper Star Couplers with up to 100 Fibre Channels," Electronic Letters, Vol. 15, No. 14, pp. 274-275, 1978.
18. Williams, J.C., and McDuffee, F.T., "An Engineering Guide to Couplers," Fiberoptic Technology, pp. 129-134, October 1981.
19. Graham, D.J., private communication.
20. MacLean, J.R. and R. W. Uhlhorn, "Physical Layer Considerations for a High Throughput, Fiber Optic Serial High Speed Data Bus," Proceedings IEEE 1986 National Aerospace and Electronics Conference, NAECON 1986, IEEE 86CH2307-7, Vol. 1, pp. 146-150, May 1986.
21. Smith, R.G. and S.D. Personick, "Receiver Design for Optical Fiber Communication Systems," Chapter 4 in Vol. 39, Topics in Applied Physics, Semiconductor Devices for Optical Communication, H. Kressel, Editor, Springer-Verlag, New York, 1982.
22. Uhlhorn, R.W. "A Robust Fiber Optic Active Star Coupler for the SAE Linear Token-Passing Multiplex Data Bus," presented at the National Aerospace and Electronics Conference, NAECON 1988, Dayton, OH., May 1988.

ACKNOWLEDGEMENT

The author wishes to thank Mr. John Ostgaard of the U.S. Air Force Avionics Laboratory, and AGARD, for the opportunity to participate in this lecture series. All original work reported in this paper was sponsored by the Harris Corporation Independent Research and Development program. Messrs. F.T. Couey, J.B. Pearce, P.A. Winkelman, and P.J. Rossana of the Harris Corporation have been particularly helpful and encouraging while preparing for the series. I would like to thank Mr. J.C. Snow for permission to use his description of the parallel internal bus.

HIGH SPEED PARALLEL PROCESSING NETWORKS FOR ADVANCED ARCHITECTURES

D. REED MORGAN

U.S. AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AVIONICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OHIO, U.S.A. 45433

SUMMARY

Many force multiplier improvements in vehicle control, situation awareness and crew decision aiding will be made possible if affordable, flyable supercomputers and associated software can be developed for next-generation military aircraft. New functional architectures will emerge because dramatic improvements in processing speed can be implemented through tightly coupled networks. Unconstrained system architectures can be developed where the system designer will have the capability to fuse together needed logical functions irrespective of previous boundaries. In addition, increased local processing (e.g., "smart" sensors) will be made possible to improve threat and target classification. Robust use of real-time artificial intelligence at both local, functional and system levels will be achievable, along with improvements in fault tolerance and system diagnostics.

Avionics supercomputer networks are expected to be used by the early 21st century to implement metafunction integration. For example, an integrated vehicle control system metafunction will be made possible. Several previously separate functions such as propulsion controls (engine and thrust vectoring), flight/guidance controls, navigation equipment, fire control and weapons control can be closely tied together to accomplish optimum maneuvers for improved survivability, fuel economy, weapon delivery effectiveness and/or fault tolerance. Other metafunctions may evolve within/across the electro-optical sensor spectrum, the radio frequency spectrum, the pilot-vehicle interface function and system-level decision aiding using machine intelligence (e.g., tactics and real-time mission planning).

Computer speed improvements of the order of 100-1000 times that of the fastest avionic compatible processors available today is needed for data, signal and artificial intelligence processing. Only parallel (i.e., concurrent) processing can potentially achieve these needed speed improvements.

This paper describes various parallel processing architecture network that are candidates for eventual airborne use. An attempt at projecting which type of network is suitable or optimum for specific metafunction or stand-alone applications is made. However, specific algorithms will need to be developed and bench marks executed before firm conclusions can be drawn. Also, a conceptual projection of how these processors can be built in small, flyable units through the use of wafer scale integration is offered. The use of the PAVE PILLAR system architecture to provide system level support for these tightly coupled networks is described.

The basic conclusions to be drawn from this paper are: (1) extremely high processing speeds implemented in flyable hardware is possible through parallel processing networks if development programs are pursued, (2) dramatic speed enhancements through parallel processing requires an excellent match between the algorithm and computer network architecture, (3) matching several high speed parallel oriented algorithms across the aircraft system to a limited set of hardware modules may be the most cost effective approach to achieving speed enhancements, and (4) software development tools and improved operating systems will need to be developed to support efficient parallel processor utilization.

INTRODUCTION

The motivation to consider the use of parallel processing can be grasped by pondering the operation of the brain. Dr W. Daniel Hillis, inventor of the Connection Machine (a massively parallel network processor) has phrased the question as follows: "how can the brain take an unstructured, complex problem and deduce the answer in a fraction of a second, when a "smart" computer may take hours or days to perform the operation? Why is the computer so stupid when it could have 10⁹ more switch changes per second than the brain?" Dr Hillis suggests that the answer lies with the brains massively parallel operation where an array of self-organizing, fault tolerant neurons perform possibly 100 steps of processing actions. [1] The concept of concurrently using many processors to solve complex problems in a "divide and conquer" strategy appears to be the only approach for real time operation. Researchers are currently attempting to partially emulate the operation of the brain in an exciting new field called neural networks and someday, adaptable learning systems may be possible. However, we must generally be content in working with processors having much larger grain size than the neuron. As will be seen, the choice of the grain size depends on many factors including: (a) the degree to which the algorithm can be parallelized (b) the complexity of the resulting network and switches (c) software control complexities and (d) fault tolerance of the resulting network.

Impressive processing speed improvements have been recently achieved through the use of Very High Speed Integrated Circuit (VHSIC) technology. For example, a 3 million instructions per second (DAIS mix) processor with 256K words of memory in about a 6"x6"x.6 (15cm x 15cm x 1.5cm) package has been built under the PAVE PILLAR program at WPAFB, Ohio. Faster processors (approximately 5 times) will be made possible by the second phase of the VHSIC program. During the early/mid 1990s, we will be reaching speed limits on how fast a given microcircuit can process data due to inherent limitations on the feature size of chip componentry. And although these incredible speeds will be exploited across an array of military systems, dramatically faster (e.g., 100 times) processing will still be required for certain applications. If these algorithms were implemented today, they could only be investigated on the ground since a large supercomputer would be required. If we limit ourselves to a smaller class of avionic problems and are able to partition the algorithms properly, the possibility exists that we can achieve the needed speeds in flyable configurations by using an array of processors implemented through advanced microcircuitry.

Several government sponsored and commercial ventures have led to the development of ground-based parallel networks that have been demonstrated as operating faster than classical supercomputers. However, these machines are still much too large for airborne use and, invariably, their speed superiority against say the CRAY X-MP (Cray is a trademark of Cray Research Inc., Minneapolis, Minn., USA.) is limited to one or a few special algorithms which uniquely match its special network architecture.

Although the concept of exploiting parallelism appears to be the way we should proceed for high speed airborne processing, two radical design features will need to be incorporated to achieve affordable and flyable hardware. These design features are: (1) processors, memories and input/output functions will need to be implemented in microcircuitry at the wafer level in order to achieve needed size reduction. This capability appears achievable by the mid 1990s, based on current research. Using wafer scale integration, extremely powerful (and highly reliable) circuits can be built using several high speed processing cells interconnected on a 4" (around 10 cm) wafer. These wafers can then be stacked, achieving a "three dimensional" computer (advanced cooling techniques may be required), (2) the high cost of both hardware and software associated with parallel machines implies that we must avoid unnecessary proliferation of basic processor architectures. The goal would be to develop a family of common, modular wafers for replicated use, with advanced CAD/CAM techniques employed to reduce cost. However, we must "personalize" the network architecture to match the parallelized algorithm. Therefore, either the network topology of wafer-based parallel processors need to be programmable, or a limited number of different (modular) wafers will need to be mixed together in the stack in order to achieve the desired speeds. Both techniques may have to be used for some applications. An exploratory development study is underway within the Avionics Laboratory to identify the best design approaches.

Several different parallel implementations are currently available to permit designers to match applications with parallel architectures. Usually, the designer will be required to trade off the complexity (and cost, weight and power) of the communication topology between the processor nodes and the cost of replicating the processor nodes. For example, the designer might choose a small number of "large grained" nodes (having expensive processors and memories) but possessing a low-cost interconnection network. On the other hand, he might want to investigate the use of a robust set of "fine grained" (having low cost processors and memories) and a costly data distribution network. In most cases however, the characteristics of the application algorithm will be the major determining factor in selecting major architectural features.

New software tools and languages are now being developed to support parallel network development. These tools include compilers that parallelize code to affect a more efficient architectural match before loading the program. Also, network simulators are being developed that aid the programmer in understanding and debugging the code. However, the field of parallel processing is so dynamic that often, immature tools and operating systems lead to frustration and disappointing results (hastily designed hardware and/or operating systems can lead to systems that run slower as more processors are added to the network).

It is important that we collectively assess the applicability of parallel processing to military aviation and help direct its development to achieve performance advancements needed in the early 21st century.

BACKGROUND

Before describing the array of architectural choices and their relative application benefits, let us first look at why the classical Von Neumann machine does not inherently support high speed processing. Figure 1 shows a simplified version of such a processor. Note that the central processing unit (CPU) is the single locus of control and that it must sequentially (serially) fetch data and programs out of memory in a "single thought at a time" mode of operation. Note that as a consequence of this design, the overall process is slowed down because: (a) control or data signals from elsewhere in the system can interrupt CPU operation, (2) the CPU experiences a "bottleneck" in attempting to accomplish processing since much of the time is spent retrieving and storing data from a relatively small percentage of total memory (i.e., only 1-2% is nominally in use at one time).

Although faster microcircuitry can be used to speed up the process, a practical limit on processing speed is eventually reached. For example, Figure 2 shows the growth in speed for airborne uniprocessors over the last decade and a projection of future growth. Figure 2 shows the speed achieved in using MIL-STD-1750A instruction set computers executing the Digital Avionics Information System (DAIS) instruction mix. Note that as the minimum feature size of the microcircuit has been reduced over time, the use of increased clock rates result in higher processor throughputs. However, a practical limit on feature size (at around 0.25 - 0.5 micron) leads to an attendant limit on clock rate (around 100 MHz), forcing the computer architect to resort to Reduced Instruction Set Computers (RISC) in order to achieve further speed gains (i.e., only simple instructions are implemented in hardware with RISC machines - they would not be efficiently used for algorithms requiring extensive complex operations such as floating point). In order to achieve higher speeds, the designer can choose a large number of simple "fine grained" machines on one end of the spectrum or a smaller number of "coarse grain" machines at the other end, and accomplish parallel processing. All avoidable "chokepoints" or processes that inhibit speed must be designed out of the system. For example, attempting to use the same memory with several processors causes a memory access bottleneck. We must design parallel systems that inhibit unnecessary I/O interrupts, minimize and remote the control of the process and distribute the memory. And this must be done in such a way to allow for fault tolerant operation.

Before resorting to the use of parallel processing to achieve the needed speed, the designer should consider the applicability of various single processor speed-up techniques. These techniques are aimed at causing a larger percentage of CPU circuits to be active at any given time. (This is a form of "virtual" parallel processing in that concurrency of operation is achieved). One approach is to use a technique called memory interleaving (if appropriate to the application). Here, large arrays of numbers are stored in separately accessible memory units (say 8-16), permitting the CPU to simultaneously access each unit over multiple lines. Further, a speed-up concept called pipelining may prove attractive.

Here, the calculations involved depend on the need to perform several small, yet different steps allowing a section of the CPU to perform part of the calculation, pass it on for the next independent step (i.e., pipeline), while simultaneously bringing in new data for processing. Use of this technique to accomplish vector multiplication of state values is common. Finally, the use of very long word processing (up to 1024 bits/word) is a means to speed up computing by simultaneously executing instructions on the data. However, care must be exercised in scheduling the operations to avoid data dependencies.

MULTIPROCESSING

In order to achieve higher speeds, the designer must eventually resort to using multiple computers. Figure 3 shows the range of "speed-up factors" to be expected as a function of the number of processors in the network. Note that chokepoints can still exist. For example, the lower curve shows the effects of memory access and system control bottlenecks when several processors attempt to share a common memory (this is not considered to be a true parallel processor configuration). Figure 3 also shows the ideal case using parallel processing where no bottlenecks occur and a perfect algorithm-to-architecture match has occurred. In between these two boundaries lies the practical, real-world processor speed up that can be achieved. Hopefully, about 85-90% of the ideal speed up can be achieved. The shape of the curve will be primarily determined by the relative number of I/O interrupts across the network, the length of time required to transmit messages between processors and the extent of memory segmentation.

Before describing the operation of various parallel machines, it is worthwhile to remind ourselves of two basic considerations: (1) the overall network must possess fault tolerant features for most airborne applications involving airborne guidance and control applications and (2) the system operation of the parallel processing network will likely be controlled by another computer - a Von Neumann machine.

PARALLEL PROCESSING SPEED UP

The single-most important issue in parallel processing is the equivalent speed up gained by the use of "n" processors. In general, a speed up of "n" cannot be expected since real world problems are not perfectly parallel (i.e., requiring an algorithm that can be executed on multiple processors without communication between these processors).

In general, the speed up factor S_n is defined as $S_n = \frac{T_1}{T_n}$ where T_1 is the algorithm execution time

required for one processor and T_n is the execution time for n processors. If λ represents the fraction of work that can be processed in parallel, and we assume that at any instant, either only one machine is working or all n of them are working, $S(n, \lambda) = \frac{1}{(1-\lambda) + \lambda/n} \leq \frac{1}{\lambda}$ for all n, λ .

Here, the execution time of one processor is normalized to unity. [3]

Note that the first term in the denominator represents the fraction of the algorithm that cannot be parallelized and the second term is the time required to execute the portion that is parallelizable. Figure 4 shows a plot of the speed up factor S as a function of λ . Figure 4 points out a fundamental property of parallel computation: unless λ is greater than 0.9, parallel networks will not provide a significant speed increase. It is imperative that the algorithm be investigated for parallelism before proceeding to parallel network implementation. Since the slope of the curves in Figure 4 vary as a quadratic relationship with n as λ approaches 1, new algorithms emphasizing parallelism and/or the use of compiler tools that reformat an existing algorithm into its most robust parallel form may be mandatory. Given that an algorithm lends itself to parallelism, then its significantly parallel portions must be efficiently mapped onto the proper network architecture if "real-time" processing of complex algorithms is to be achieved. The architecture selected must exhibit: (a) the proper "grain size" for each processing element, (b) highly efficient communications between processing elements, (c) high speed memory accessibility (particularly for large sized applications), (d) good load balancing for the processing tasks over time and (e) real time, fault-tolerant operation.

In summary, an efficient parallel processing network must possess a processor interconnect topology and system control strategy that grows weakly with increasing n. Examples of various parallel network topologies are presented below.

TERMS USED IN PARALLEL PROCESSING

Before proceeding further, a few terms need to be defined to aid information exchange.

Perfectly Parallel: an algorithm/parallel network application that is executed on multiple processors without communication between the processors [2] (Perfectly parallel systems are an artificial construct for comparison with other meaningful, "real world" algorithm/network realizations).

Explicitly Parallel: an algorithm/parallel network that can be executed on multiple processors only with communication with neighboring processors. [5] (nearest neighbor processing is particularly useful in comparing parameters for fine grain network applications).

Host Processor: a separate, Von Neumann-based machine used to control the operation of the parallel network (e.g., synchronization of nodes, data processor routing, control/distribution of data from external sensors, etc.).

Node: a processor/memory pair in the network. The node usually contains an input/output section that receives incoming data and routes messages and data to other nodes and could conceivably contain a co-processor for special purpose applications. The associated memory may be directly connected to the local processor or allowed to be jointly addressed by the local processor and other nodes in the network.

SIMD (Single Instructions Multiple Data): a single instruction stream network control concept where all processors are controlled (by the host) by broadcasting sequential single instructions to each node. Typically, each processor has the option of executing each instruction or ignoring it, depending on the internal state of the processor.

MIMD (Multiple Instruction Multiple Data): a multiple instruction stream network consisting of a collection of connected autonomous nodes, each capable of executing their own programs. A loose mechanism for synchronizing operations between nodes usually is employed.

Note: It is possible to use a SIMD and MIMD network to emulate the other with various degrees of complexity and overhead software for control. Either network approach can be used for virtually any application - differences in performance, cost and complexity must be addressed however.

Availability of Prototype & Commercial Parallel Processors

In order to obtain insight into the types of parallel processing machines which need to be investigated for airborne applications, see Table 1 which shows a partial list of various machines. (The total number of parallel processor models developed will likely reach 100 by 1990 with over 30 vendors). Referring to Table 1, it can be seen that vendors have chosen to implement these machines with different topologies, interconnecting vastly different numbers and types of processors. Application guidelines from the proper selection of these various network architecture features will be addressed below. The reader is reminded that these machines are designed for ground-based operation for scientific and engineering applications. Extensive size reduction must be accomplished for airborne applications.

Candidate Architectures

In discussing digital processing architectures, it is important to differentiate the system architecture used to integrate aircraft functional elements and the parallel processing network architecture.

System-Level Topology

The PAVE PILLAR System architecture provides a physical and logical means of integrating and controlling functions. Its high speed data bus, token passing protocol and the Ada-based operating system allows the designer great flexibility in integrating processing resources and sensors (e.g., flight control system), signal processing functions (e.g., coupling of common signal processors for fault tolerance purposes) and the overall system-wide integration of flight control, avionics, weapons, sensors, displays, etc. The reader will note, from Figure 5, that PAVE PILLAR is a parallel processing system of computing resources communicating over a bus-driven interconnection network. It is not generally appreciated that military avionic architectures have been supporting parallel network operation since the mid-1970s (the author is aware of the DAIS system being introduced in this time frame). The motivation behind such networks was driven by the absolute requirement for real-time computation (hence, partitioning functions to different computers). However, the already established avionic engineering disciplines and functional partitioning necessitated by the earlier use of hard-wired, analog-based avionics further ensured that separate functionally-driven processing would be used when more flexible digital processing and multiplexing were introduced in aircraft.

Multiplex bus-driven architectures possess the best features for integrating loosely coupled networks of processors (i.e., ease of integration/retrofit, fault tolerance, ease of system control). Time delays such as bus latencies caused by bus control protocols or processor interrupts by executive control are usually not significant problems for this class of applications.

However, once we must resort to accomplishing high speed domain-based parallel processing (i.e., within a specific function) that requires many tightly coupled processors to accomplish the task in real time, a bus oriented design may be found to be inadequate for inter-node connectivity. Notice however, that the bus approach will still be useful in integrating separate parallel processing networks. Referring again to Figure 5, the reader can visualize that the basic PAVE PILLAR architecture could potentially support the use of tightly coupled parallel networks at virtually every possible level. For example, such a network could be used within a particular signal processor to perform special signal analysis or within a sensor or weapon. Further, a parallel network responsible for high level system management functions could be connected to the mission avionics bus. Finally, we could consider using a combination of parallel networks and PAVE PILLAR networks or buses to create higher levels of more closely coupled functions.

Why do we need dramatically higher speeds on future aircraft?

Future aircraft processing avionics systems must provide dramatic performance improvements in two basic areas:

- (1) Vehicle system control integration to achieve improved control, energy maneuverability and fault tolerance for enhancements in survivability and weapons delivery.

This will require the coupling/coordination/integration of engine controls, thrust vectoring (where applicable), flight control, navigation and guidance functions, fire control and weapons control. With this propulsion/flight/guidance/fire control synergy, aircraft should be able to perform more effective maneuvers to evade missiles, out-maneuver enemy aircraft for favorable attack conditions, improve survivability against heavily defended ground targets through maneuvering attack, minimize fuel consumption, etc.

Such an integrated system will require a robust use of high speed, fault tolerant networks to perform artificial intelligence reasoning for optimum control and reconfiguration strategies and to manipulate large arrays of matrices describing control coupling effects.

(2) Improved situation awareness of enemy and friendly forces in complex, dense and rapidly unfolding environments will require the use of multispectral sensor integration and coordination, automatic target and threat recognition, development of pilot decision aiding tactics and recommendations, reduction of excessive data for presentation and the generation of distilled information. These processes in turn require orders of magnitude improvements in signal and image processing speeds and artificial intelligence processing.

In achieving both of the above capabilities, we can expect the pervasive use of machine intelligence from the sensor level up through the system level. Because of the need to fuse and control functional information, we can expect a trend towards integrated metafunctions (i.e., collection of functions). For example, the integrated vehicle control function described above may be eventually implemented as a closely integrated set of vehicle and trajectory control functions. Further, the merger of electronic warfare, communications and radar functions may evolve into an integrated Radio Frequency (RF) metafunction. At the highest system level, a decision aiding metafunction to recommend overall tactics and mission plans may emerge. Advanced, high speed parallel networks will be needed to not only provide the necessary sensor-level processing within the metafunction, but the network will likely be used to integrate the separate functions within the metafunction. It is the author's view that the metafunction processing sites would in turn be integrated through a high speed bus structured network (i.e., PAVE PILLAR).

PARALLEL NETWORK CANDIDATE TOPOLOGIES

This section addresses the various nodal interconnection techniques that the designer may wish to consider. A general description of several network topologies will be presented, along with a discussion of currently available networks that reflect that topology. Although the examples will only touch on a portion of the available networks, they are representative of the choices presented to the designer. The failure of most parallel processing algorithms to scale properly for applications requiring more than a few processors is usually attributed to the glut of communications (i.e., the interconnection scheme was poorly designed). [4]

CIRCUIT SWITCHED NETWORK TOPOLOGIES

The circuit switched network is the simplest approach to interconnecting nodes if their number is relatively small (e.g., under 20) and inter-nodal communication is not erratic. Figure 6 shows a circuit switch approach that has non-adaptive message/data routing (i.e., the path of the message between nodes depends solely on its source and destination). This type of switch is similar to telephone network where a node-node link is established and held as long as needed. This approach simplifies control problems (e.g., simple software), but could be inefficient if the applications requires highly dynamic message passage.

This type of switch is used by the common signal processor (CSP) as its switching architecture. (Figure 6) Note that no real restrictions have been placed on the nodes in Figure 6. There could be floating point processing elements, memory elements, a fine grained parallel processor network, a data processor or a combination of these units. In fact, a major attribute to the circuit switched approach is its flexibility and simplicity in accommodating different kinds of nodal processing that need to be integrated. The CSP design allows the ultimate in hardware flexibility by building common nodal modules (including the switch itself) for easy insertion into a common interconnect backplane. However, the main limitation of the circuit switch is that the complexity (weight, volume, etc.) grows as the square of the number of nodes. Hence, doubling the number of processor nodes leads to four times the number of switched ports. Therefore, for large networks, this technique becomes unacceptable.

PACKET SWITCHED TOPOLOGIES

Packet switching allows nodes to communicate through addresses, where new routes are established each time a line is established (i.e., similar to a post office network). These networks can be used for shared memory access (i.e., to permit one processor node with its local memory to access any other memory in the network) or to transmit data or commands to other nodes.

Shared Memory Switching

Shared memory switching is a subset of the packet switching topology and supports applications where there is a random distribution of data which requires access by individual nodes. Figure 7 shows a simplified example of a small (limited) access shared memory switch.

The features of such a network can be better understood through a description of the Butterfly Parallel Processor. ("Butterfly" is a trademark of Bolt, Beranek and Newman, Inc., Cambridge, MA, USA).

Collectively, the memory of the nodes forms the memory of the entire machine, with the Butterfly switch to used make remote accesses. Typically, local memory referencing within a node takes about 2 microseconds whereas remote accessing may take 5-6 microseconds. The system is currently configured with up to 256 (MIMD) processor nodes, each consisting of Motorola 68000 chip, 1-4 megabytes of memory, a memory management unit, and a processor node controller. Each node is capable of executing 500,000 instructions per second. The bandwidth through each processor path is 32 megabits/sec - thus a 256 node machine has a raw processing power of 128 MIPS, 256 - 1024 Mbytes of memory and an interprocessor communication capacity of 8 gigabits/sec. Because there is not a serious performance penalty to access remote data, the user can organize data without serious concern over where it is located. Nearly linear speedup has been observed for large complex matrix manipulations and large configurations are being used for image processing and computer vision. [5]

The uniqueness to the Butterfly stems from its switch (often referred to as a Banyan switch) which allows every node to link together. The number of switches in the network only increases as $n \log_2 n$ (as

compared with n^2 for the crossbar). For a 100 node system, the Butterfly switches would require about 500 wires, compared to 5000 for a cross bar system.

The topology of the switch is similar to that of the Fast Fourier Transform Butterfly - hence the name. A simplified view of the switch for a 16 node arrangement is shown in Figure 8. Each switch is a 4 input, 4 output unit custom chip. Each node uses packet addressing to route the packet from the source to a destination node. To send a message from node 5 to node 14, node 5 builds a packet containing the address of node 14 (1110 binary) followed by the message data. The sending switch strips off the two least significant address bits (10) from the packet and uses them to select the number 2 (10 binary) port to transmit the signal. The receiver switch strips the 11 off the address bit to switch the data packet out of its port (11 binary) to processor 14. Note the ingenious arrangement of the processors relative to the switches. Processors 2,6,10,14 (binary 0010,0110,1010,1110) all have the same "10" bits in the receiving nodes of the third switch. Similarly, the four transmitter switches are connected to the third receiver by separate "10" lines. Any switch can issue the "1110" packet and be linked to processor 14. These switch units can be cascaded both horizontally and vertically for larger networks. Because there is a not a dedicated path between every nodal pair, contention can occur where two messages can reach the same node. When this occurs, one message is allowed to pass with the other being retransmitted. Redundant paths are implemented for fault tolerance. A host processor is used to edit, link and compile data and programs. [5]

HYPERCUBE TOPOLOGY

The hypercube topology is potentially suited for applications requiring a few nodes to thousands of nodes. Each node operates as an independent unit which can communicate while processing. Each node has its own memory and a copy of its own operating systems. Its chief benefit may lie with its significant flexibility since it is a superset of other network topologies. Grid, ring and tree networks can be configured by programming the control host computer. A 64 node processor hypercube has been clocked at twice the rate of a Cray X-MP (although the average speed is only one-fourth the speed of the CRAY). [6] Hypercubes use a message passing approach that allows nearest neighbor communications from one node to the next until the destination node is finally reached (with intermediate nodes thereby passing on the message). The dimension, or order of the hypercube is defined as the maximum number of inter-nodal links throughout the entire cube. Figure 9 shows the configuration of various dimensions. For example, if the dimension is 3 (i.e., third order hypercube), a maximum of three message links are required (average of 1.5) and the number of nodes being $2^3 = 8$. If d represents the cubes dimension, or order:

d = max # of links

2^d = number of nodes

$\frac{d}{2}$ = time average of number of links (assuming no contentions)

$n2^d$ = number of (two way) connection paths between nodes

For example, a 128 node hypercube would be of order 7 and require $7 \times 128 = 896$ communication channels (the programmer needs only to specify the nodes, not the paths taken, however).

The power of the hypercube, in addition to its topology flexibility, lies with the fact that the number of message links varies linearly with d , yet the number of total processors (a measure of total system speed) goes up as 2^d . Thus, a six cube only has one half less "hop" on average than a 7 cube but the six cube has one-half the number of nodes. However, the large number of resulting interconnections, as the order gets larger, can prove to be the limiting design factor. These interconnections could account for the majority of circuit complexity, weight and system unreliability and can result in data latencies due to communication protocol. A goal of hypercube design is to achieve a ratio of computation time to communication time of about 10. Load balancing may also prove a design problem for certain problems. [2]

The Intel Hypercube from Intel Scientific Computers, Inc., of Beaverton, Oregon, USA, consists of two functional elements - the cube and the cube host processor. Units containing 32, 68, or 128 nodes are available, each node using a Motorola 80286 processor and 512 kilobytes of local memory.

The Connection Machine (Trademark of the Thinking Machines, Inc., Cambridge, MA, USA) however, represents one of the most robust forms of parallel processing. This design, shown in Figure 10, consists of a massively parallel SIMD network having a maximum of 65,536 nodes. Each node consists of 4096 bits of memory, a message router circuit and a one bit processing unit. The host can control individual nodes or collections of nodes to achieve virtually any type of calculation. Algorithms such as image processing or semantic nets need a large array of small processors and are particularly suited for this architecture class. The current Connection Machine is configured in a 1.4x1.4x1.6 meter package. It is capable of one billion instructions per second (BIPS) (for a 32 bit addition benchmark) and has demonstrated 6-7 RUP performance in information retrieval applications. This machine has a 12th order hypercube topology. The Connection Machine can be programmed to emulate a MIMD structure but its relative efficiency is unknown. Also, since each node can be used to test neighboring processor and message routers, a fault tolerant communication scheme is possible. However, the resulting complexity of this approach is also unknown. [1]

SYSTOLIC ARRAYS

Systolic arrays capitalize on the regular structure of certain algorithms. The term "systolic" refers to the pipelined computations that are scheduled for reuse as it moves through the array and draws on the comparison of the human circulatory system (i.e., the heart sends blood through the circulatory system by frequent systolic pumping of small amounts of blood). Systolic arrays can be considered as a cost

effective approach for algorithms having a large input (or output) bandwidth. Inputs enter the array under the overall control of a host computer (see Figure 11). The data is sequentially acted on in a periodic SIMD manner as it moves throughout an array or a series of processing elements. Although significant speeds have been achieved using only a modest amount of hardware (for properly matched algorithms), debate continues as to the general purpose nature applicability of systolic arrays. It appears particularly suitable for high speed image processing, signal processing and for matrix arithmetic. It appears that fault tolerance can be achieved through the use of backup sparing. [7] However, concern exists as to whether near real time reconfiguration can take place. [8]

OBSERVATIONS ON SELECTING PARALLEL NETWORKS

The cost effective choice of using a SIMD or MIMD network architecture depends on the characteristics of the application.

In general, the SIMD-based network is preferable when the algorithm is well structured, and has a regular pattern of control for the cooperating nodes. If the algorithm displays these properties, the network control hardware that issues the highly synchronized single instruction stream can be shared by all nodes, which also will result in simpler software programming. Example algorithms that generally possess these properties include matrix calculations, certain types of artificial intelligence applications (e.g., semantic networks), image processing and signal processing.

A MIMD network architecture will likely be preferred if the algorithm control flow is highly complex and data dependent. For such algorithms, much of the SIMD network nodes would sit idle much of the time waiting for the proper instruction.

Example MIMD-oriented algorithms include certain types of artificial intelligence applications (e.g., rule based knowledge acquisition), for cooperating expert systems (e.g., tactics planning, mission planning, system health management, situation assessment) and executive/system control of complex processing sites.

It must also be observed that a mixture of the SIMD and MIMD architectures may be the preferred design for some applications.

Finally, the designer must also address the type of computer architecture needed at the node. For symbolic oriented (i.e., AI processing), a simple instruction set computer architecture is needed. For numeric based algorithms, complex floating point and vector processor features will be needed.

OBSERVATIONS ON PROGRAMMING PARALLEL PROCESSORS

Although some parallelizing compilers and some software development tools are available, the programmer of parallel computers must still be extremely careful in matching the structures of the algorithm and architecture. New languages and more extensive tool sets are being developed, but until they are matured and prove themselves, the programmer must remain knowledgeable of the machine operation and have intimate familiarity with the algorithm. The programmer must carefully divide the problem into logically separate pieces that will execute concurrently and then determine how the processors should communicate to exchange data and messages. Hopefully, the programmer will be able to choose from a variety of machines. This section provides some gross observations about the tradeoffs needed. Many of these observations are taken from Reference 9.

The programmer needs to first understand the size and number of the iterative loops in the algorithms for purposes of decomposition. This process enables the programmer to determine the needed granularity properties of the network (i.e., granularity is a measure of the extent each node spends in processing versus communication between nodes).

Coarse grained nodes would be selected if relatively long, independent processing segments are typical of the calculations. Note however, that the overall network could still be architected as a "nodes of nodes" if fine grained operations were required within a "coarse grained" loop. This concept is shown in Figure 12 which shows a generalized network of parallel machines.

If a fine grained application is identified, fewer instructions will be executed between inter-nodal messages.

At opposite ends of the spectrum, a large grained algorithm might be implemented by a few coarse grained computers having relatively slow communication links or a small grained algorithm by a large number of small nodes requiring extensive and fast communication links.

Once the programmer determines the algorithm granularity, he then must select the best type of inter-nodal communication. This selection usually involves one of three approaches: (1) a bus structured approach (for a limited number of nodes), (2) shared memory or (3) message passing. The first option will not be discussed here due to its relative high exposure in general avionics (e.g., DAIS, PAVE PILLAR bus structures). Shared memory networks allow data stored in a shared memory location to be read by all other nodes. This approach requires synchronization of the nodes for purposes of writing, reading and posting data. The reading processor is then allowed to access information with minimal operating system intervention. For a message passing approach, the specified receiver requests transmitted data from a specified sender and waits for its reception. This approach has implicit system synchronization, potentially allowing simpler programming. Message passing overhead may become excessive and needs to be analyzed first, possibly through simulation.

However, the whole question of selecting the proper communications network is also clouded by the issue of load balancing, which translates to an issue of complexity for the host computer software and grain size. For example, if the algorithm is partitioned into modules having fixed but different processing

complexities, processors will be left idle waiting for others to finish. To balance the load, one could go to smaller modules (if possible), leading to more nodes (less efficiency due to inter-node communication). Thus, a conflict occurs requiring a tradeoff of communication link cost and size versus ease of programming.

For parallel programs that employ shared memory techniques, the subtasks tend to be more tightly coupled than with message passing techniques, and frequent inter-node communication is expected. Dynamic load balancing is efficient for relatively small tasks. In using message passing, the programmer will attempt to reduce interprocessor communication by statically partitioning subtasks close together even at the expense of load imbalances. Rigidly structured applications not requiring tight subtask coupling would therefore benefit by the message-passing approach result in efficient communications [9]. As can be seen by the above discussions, matching the algorithm to the architecture to achieve efficiency while minimizing programming complexity and achieving a hardware design that is affordable, will often involve a complex decision process. A significant investment in computerized simulation tools will be needed to intelligently affect the best (compromised) overall design.

MILITARY AIRBORNE APPLICATIONS OF PARALLEL NETWORKS

As threat capability increases, crew members are forced to operate under more time compressed, stressful situations where they are faced with a vast array of data relating to threat, terrain, aircraft and weapons. The crew is required to develop information from the data, assess the situation and act quickly in life-threatening situations, while continuing to fly the aircraft. Dramatic improvements are needed to help the aircrew understand the "big picture" through improved situation awareness from sensors and displays and more automation must be used to simplify the execution of his intent. Improving situation awareness and simplifying the control over the weapon system will require the robust use of: (a) "smart sensors" that permit the automatic recognition of threats and targets, (b) selective fusion of sensory information to improve confidence in threat/target recognition, particularly in dense EW environments, (c) coupling of vehicle functions to improve performance and fault tolerance, and selective automation of guidance and control functions, (d) crew decision aiding, and (e) simplified or distilled presentation of only significant information to the aircrew. Achieving these capabilities will require extremely high throughput processing throughout the weapon system for real time signal processing, heuristic processing and data processing. In some instances, high speed uniprocessors or a small number of bus-connected processors will be adequate. In other areas, we must resort to parallel networks to achieve the needed speeds. The following discussion provides some insight into the characteristics of some of these high speed applications.

SIGNAL AND IMAGE PROCESSING

Programmable signal processing is currently being utilized for EW, radar and CNI airborne applications. This capability has allowed highly versatile sensor moding, fault tolerance and flexibility.

The author expects a continued escalation of the use of avionic sensors requiring very high speed processing. Examples include, (1) real time high resolution synthetic aperture radars, (2) low ground speed moving target indication by radar, (3) automatic target recognizers, (4) versatile beam steering and shaping of RF signals for signal transmission and reception, (5) terrain following/terrain avoidance/obstacle avoidance sensing and (6) detection of signals in jamming environments. Image processing applications such as automatic target recognition can require upwards of about four billion operations per second. Upwards of 5 billion complex operations per second may be required for these applications. In general, the above signal processing applications will require matrix-based, complex, floating point arithmetic employing various kinds of signal filtering and fast Fourier transforms (FFT) operations.

Many of the above described classes of parallel architectures perform matrix calculations extremely well. For example, a 128 node Butterfly processor still has an effective processor speed up of over 115 nodes when solving a 400x400 matrix multiplication [5].

Systolic arrays are also particularly known for their efficient processor speedup capability for matrices and image detection and manipulation.

Parallel network processing will likely be used for real time graphical portrayal of overlaid information of terrain, targets, weaponry status, steering cues, vehicle health status, etc.

ARTIFICIAL INTELLIGENCE (AI)

Codifying previously acquired human knowledge and then retrieving it to assist the aircrew in time-compressed situations will certainly be useful for future military aircraft. A large community of researchers are currently developing AI algorithms for applications that touch on virtually every area of military avionics. For example, AI is likely to play a future role in developing offensive and defensive tactics, mission replanning, assessing threat intent, understanding vehicle health status and recommended reconfiguration options, selecting proper information for presentation to the aircrew, selecting gains for flight control, controlling engines, selecting appropriate sensors/parameters for fusion, speech recognition, understanding images, etc. The resulting "symbolic" processing requires simple operations such as comparison and selection, pattern matching and logical operations. Robust arithmetic operations such as floating point (needed by signal processing) are not needed, leading to the expectation that an array of "RISC-like" computer nodes (with its attendant simplicity and speed advantages) will be sufficient.

Unfortunately however, the literature appears void of examples of militarily significant real time applications that have been demonstrated to date, even using large ground-based computing networks. Only a very few AI-oriented machines have entered the marketplace (e.g., Connection Machine); however, several new AI machines are under development. [10].

Reference [10] states that the fundamental AI operation is search, demanding intensive memory access and I/O activities rather than CPU operation. Dynamic resource allocation and load balancing are important design considerations. The large memory requirement compounds search time since no "locality of reference" exists and algorithms are sometimes non-deterministic. [10] Achieving real time operation of a set of cooperating interactive AI modules is viewed as the ultimate challenge since we must somehow find a way to schedule each module, forcing it to arrive at a "reasonable" answer in time to support another module's needed input.

Parallel processing will be mandatory for many real time AI applications for future military aircraft. The reader should however not lose sight of the possibility that some isolated applications may be readily accomplished by a single high speed processor or a small ensemble of processors. Also, there are researchers who believe that improved knowledge representation and parsed knowledge bases will reduce processor speed requirements. Current estimates for the real time implementation of AI applications range from a "few million" operations per second for a relatively modest algorithm to a "few billion" operations per second for a robust, cooperative AI system. Due to a dearth of data, these estimates should only be viewed as indicative of the need for much faster computing on aircraft. Bench marks are needed to permit better speed sizing and they are expected to be available over the next few years as government, corporate and academic research efforts are reported. It is the author's view that significant work remains in establishing which architecture(s) should be used for efficient AI real time processing. It may result that the selected architecture will vary with the type of knowledge representative.

INTEGRATED VEHICLE CONTROL

Many new capabilities will result if we are able to dynamically integrate vehicle control, propulsion/engine control and fire control functions. High maneuverability around ground-based threats, improved weapon firing opportunities, missile evasion and improved vehicle control during terrain following and avoidance should result. Maneuvering flight during weapon delivery should minimize exposure and hence reduce lethality.

The question is whether high speed parallel networks will be required to affect proper integrated vehicle control in that separate controls (e.g., engine, fire control, flight controls) are adequately accommodated today through uniprocessors (multiple computers are used only for redundancy).

The author believes that three factors will drive some integrated designs towards parallel network-based systems. These factors are: (1) the control system will likely be implemented as a tiered, hierarchical system. Tiered levels include (a) system control and reconfiguration of the integrated system, (b) subsystem functional control and reconfiguration and (c) sensors and actuators with some embedded processing. More complex processing algorithms will have to be implemented and expert system algorithms will likely be used as part of the overall algorithm in detecting and isolating failures and selecting reconfiguration strategies. Even if these AI algorithms are "simple", high iteration rates (e.g., 128 times/sec) implies the need for high speed parallel processing, it is anticipated that robust matrix-based state equations will need to be solved at high rates to properly couple airframe and propulsion controls. Also, fault tolerant design considerations of such an integrated system may lead a parallel network of many "simple" machines that are partitioned to accommodate dedicated functions for failure localization and/or graceful degradation. Extensive redundancy of computing and interconnection networks will be required to achieve the needed fault tolerance.

AIRBORNE PACKAGING OPPORTUNITIES

Use of VLSI circuitry applied to wafer scale integration and leading ultimately to three dimensional, stacked wafer computers will enable processor size reduction at least as dramatic as the hand-held calculator revolution. Techniques for on-surface interconnection of processing micro-circuit cells on wafers are being perfected and techniques for interconnection stacked wafers in a "3-D" array are under development. [11]

The resulting "3-D computer" will consist of about 90% silicon (as compared with 1-15% with today's military processors). Reliability is substantially enhanced by deleting the use of printed wiring boards with its thousands of chip-to-board solder connections. Also, total system speed of the multi-cell wafer is dramatically enhanced due to the absence of high capacitance wires which inhibit chip-chip high speed data transfer using today's packaging approach.

The designer can now seriously contemplate wafer-based designs where input/output, processor, memory, control and custom cells can be placed on one wafer, or homogeneous cells can be placed on wafers, with data transfer between wafers accomplished by small metallic bridges across the array of the wafer [11] or by photonic interconnects. Simply stated, a hand-held (e.g., 10 cm x 5 cm) parallel processor capable of executing billions of operations per second is reasonable by 1995. For example, Reference [11] describes a stacked cellular array being constructed at Hughes Research Laboratories in California, USA. This SIMD parallel processor was designed for image and signal processor applications and consists of five different wafers. Very simple accumulator cells are used (32 x 32 cells on one wafer currently, with a 128 x 128 cells per wafer being constructed). About 10 billion operations per second is expected for a 15 wafer (5 different types) stack. [11]

In achieving a "hand-held supercomputer", a few basic caveats must be stated. First, there is the question of achieving high yield of the cells across the wafer. Here, the tradeoff is basically one of selecting the minimum feature size (and hence speed) of the separate cells. Heat removal problems may be encountered if each cell is driven at high clock rates. Direct immersion liquid cooling may be required for some designs. However, the side benefit of being able to achieve "chip-level" fault tolerance becomes a reality. Secondly, wafer interconnection technology will need to be improved. One exciting possibility is to use optical signal transfer between wafers (e.g., using gallium arsenide diodes mounted on the silicon wafers). Third, and always very important, is the algorithm-to-architecture matching

problem. There are two basic design choices: (1) use customized, stacked wafer processors having unique wafers for every high speed parallel network application on the aircraft. Although this approach maximizes the processing speed and reduces wafer interconnect complexity, both hardware design/manufacture/support and software development/support will be quite expensive for such a design approach, (2) use a family of wafer modules, and replicate their common use across various applications. If this approach is feasible (it currently is under investigation), the wafer interconnect structure will be more complex and processing speeds, although possibly adequate, will not be optimum. However, using CAD/CAM techniques, a limited set of relatively low cost wafers could be designed and manufactured in large quantity, thereby permitting "learning-curve", high volume cost savings to occur. Further, the concept of developing programmable topologies across the cells, thereby enabling a further dimension in architecture personalization to the algorithm is possible. Finally, a standard family approach allows consideration of a modular operating system that controls the overall modular set of wafers. The feasibility of this concept is also under investigation.

CONCLUSIONS

The potential applicability of parallel processing networks for future airborne applications is an exciting concept which will depend on the integration of several new technologies. However, it appears that the questions surrounding these machines are mostly directed towards selecting which applications will be used, which network should be used and what technologies should be used for the flyable design - there seems to be little question about ultimate feasibility. However, there will certainly be a period of "growing pains" which researchers must go through in learning what architectures work for which applications. However, complex and costly software is the biggest concern since basic limits on affordability could be reached if sufficient "front end" support software is not developed.

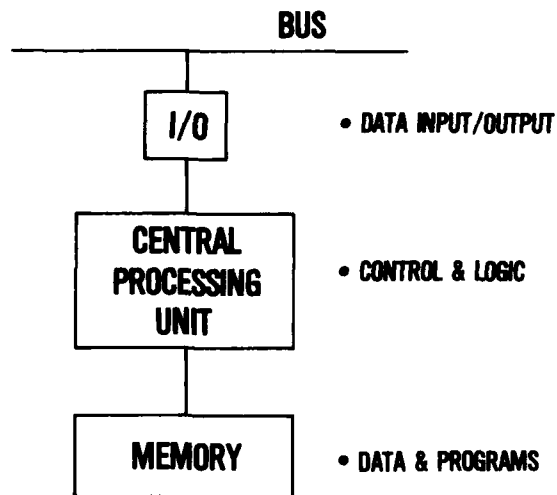
Based on on-going work in applications and technology, we should be in a good position to judge the general direction of parallel processing around 1990. By 1995-96, if proper funding is made available, flyable, cost effective parallel networks should be a reality.

BIBLIOGRAPHY

- [1] Hillis, W. Daniel, The Connection Machine, Cambridge, MA, USA and London England, U.K., MIT Press, 1987
- [2] Wiley, Paul, A Parallel Architecture Comes of Age at Last, IEEE Spectrum, Jun, 1986, pp 46-50
- [3] Buzbee, William, Parallel Processing Makes Tough Demands, Computer Design, Sep 1984, pp 137-140
- [4] Johnson, Howard, Crossbar Circuit Switch, BYTE Magazine, May 1985 [5] Bolt, Beranek and Newman Inc., Butterfly (TM) Parallel Processor Overview, Version 1, 13 Jun 1985
- [6] Bond, editor, Parallel Processing Concepts Finally Come Together in Real Systems, Computer Design, 1 Jun 1987, pp 51-74
- [7] Abraham, J. et al, Fault Tolerance Techniques for Systolic Arrays, Computer Magazine, Jul 1987
- [8] Fortes, A. Band Wah, B., Systolic Arrays - From Concept to Implementation, Computer Magazine, Jul 1987
- [9] Howe, Carl and Moxon, Bruce, How to Program Parallel Processors, IEEE Spectrum, Sep 1987
- [10] Hwang, K. et al, Computer Architectures for Artificial Intelligence, Computer Magazine, Jan 1987
- [11] Grinberg, J., et al, A Cellular VLSI Architecture, Computer Magazine, Jan 1984, pp 69-80

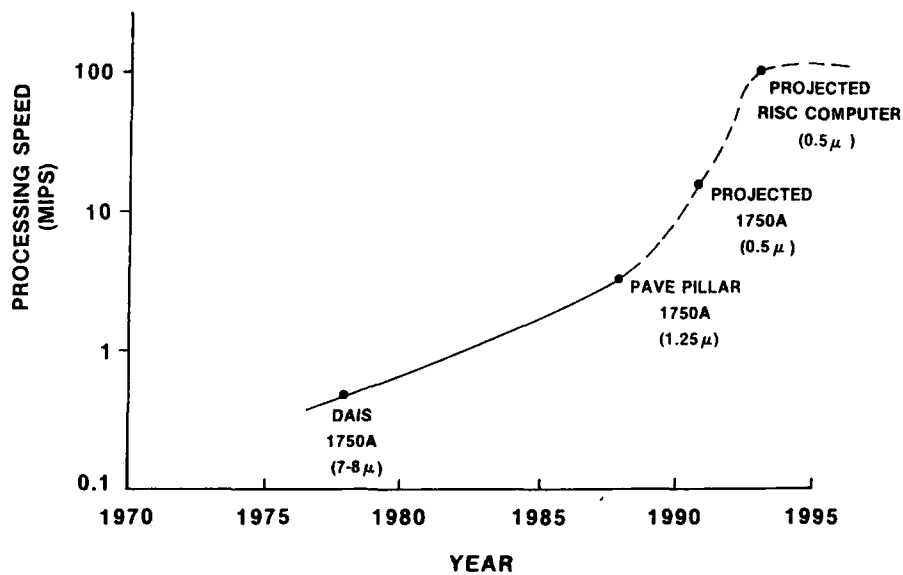
TABLE 1
EXAMPLES OF AVAILABLE PARALLEL PROCESSOR NETWORKS

| PROCESSOR NAME | MANUFACTURER/ DEVELOPER | TOPOLOGY | MAX NO. OF NODES BUILT | MAX MEM. PER NODE (Kbytes) | NODE CPU WORD LENGTH | SPEED (MIPS/NODE) |
|-------------------------------|---|------------------------|------------------------|----------------------------|----------------------|-------------------|
| COSMIC CUBE 1983 | CAL TECH, PASADENA, CA USA | HYPERCUBE | 64 | 128 | 16 | .7 |
| MARK II 1985 MARK III 1986 | JET PROPULSION LAB, PASADENA, CA, USA | HYPERCUBE HYPERCUBE | 64 1024 | 256 4K | 32 32 | 1.0 1.0 |
| IPSC (1985) IPSC (1987) | INTEL, BEAVERTON, OR, USA | HYPERCUBE HYPERCUBE | 128 128 | 512 1024 | 32 32 | 1.0 4.0 |
| CONNECTION MACHINE (1986) | THINKING MACHINES, CAMBRIDGE, MA USA | HYPERCUBE | 65536 | 0.5 | 1 | .015 |
| BUTTERFLY (1986) | BOLT, BERANEK & NEWMAN, CAMBRIDGE, MA USA | BANYON SWITCH | 256 | 1K | 32 | 1.0 |
| WARP | CARNEGIE MELLON U. PITTSBURGH, PA USA | SYSTOLIC ARRAY | 256 (10 Nom) | 16 | 32 | 10 MFLOPS |
| MASSIVELY PARALLEL PROCESSOR | GOODYEAR AKRON, OH USA | NEAREST NEIGHBOR | 16,384 | 1K | 1 | .02 MFLOPS |
| COMMON SIGNAL PROCESSOR | IBM MANASSAS, VA USA | CIRCUIT SWITCH | 24 | 4 | 32 | 125 MFLOPS |



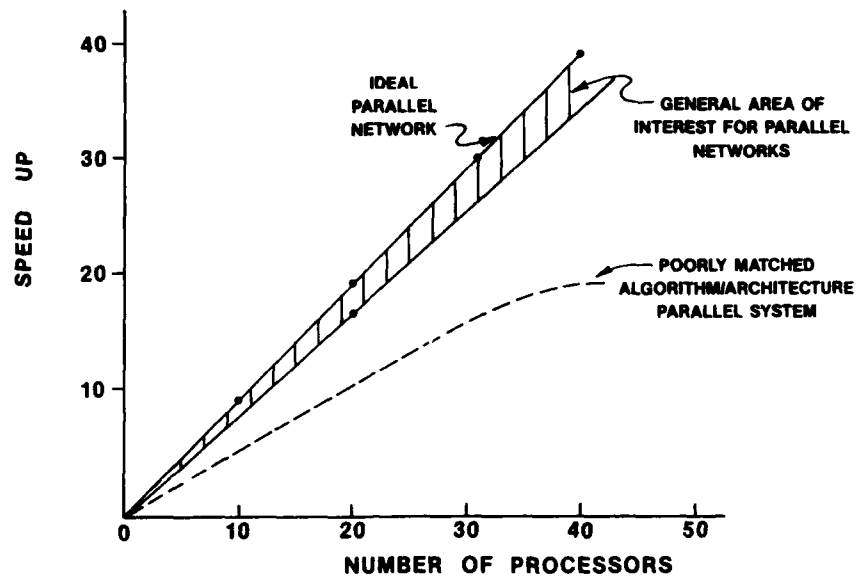
SIMPLIFIED VON NEUMANN COMPUTER ARCHITECTURE

FIGURE 1



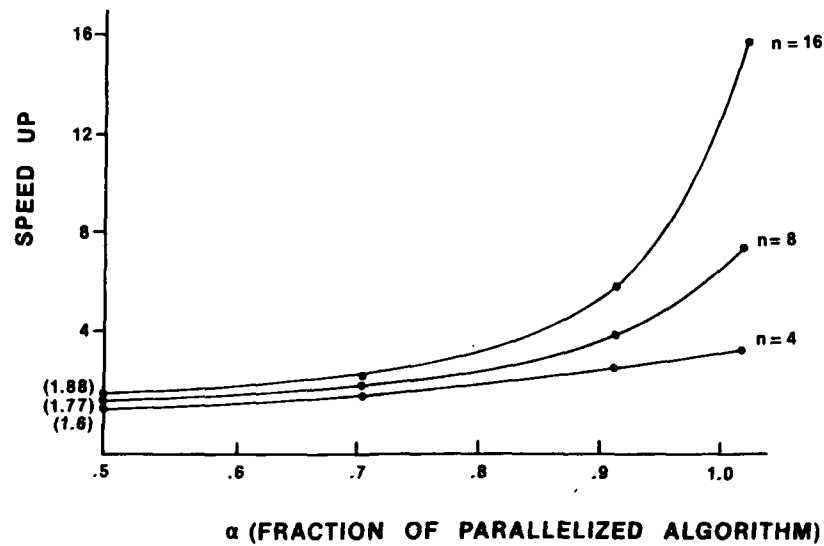
GROWTH IN PROCESSOR SPEED FOR AIRBORNE DATA PROCESSING

FIGURE 2



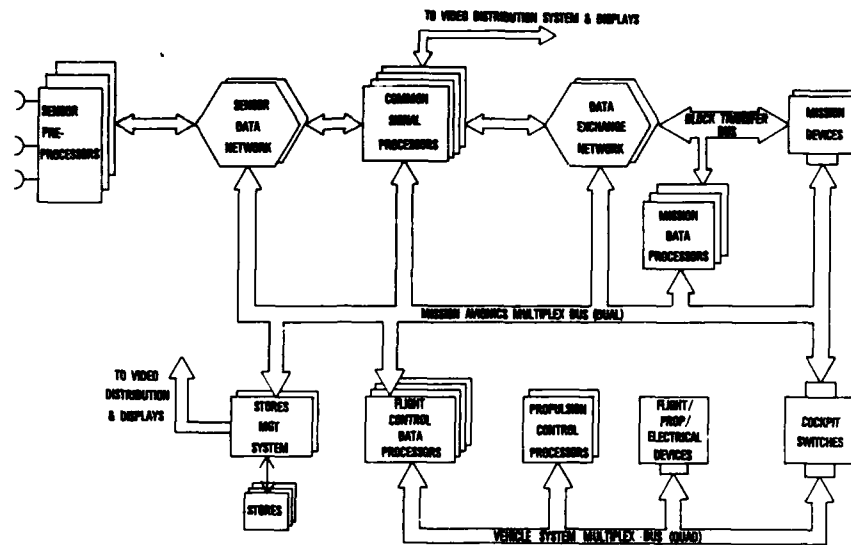
SYSTEM PROCESSING SPEEDS FOR VARIOUS MULTIPROCESSING CONFIGURATIONS

FIGURE 3



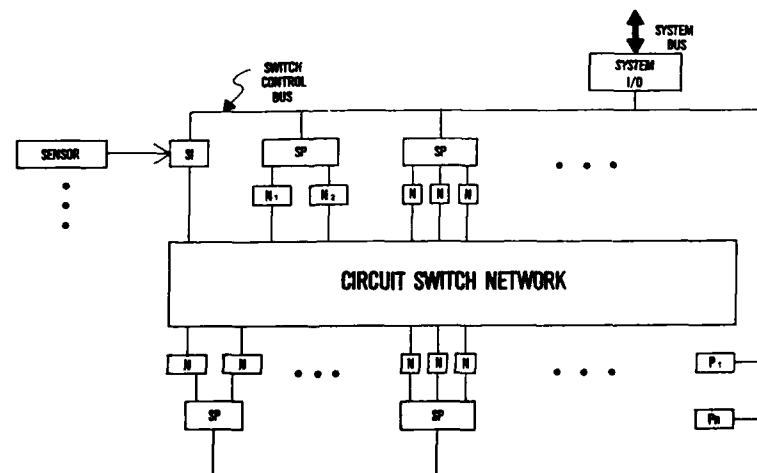
SPEED UP FACTORS FOR PARALLEL NETWORKS RELATIVE TO FRACTION OF PARALLELIZED ALGORITHM [3]

FIGURE 4



PAVE PILLAR ARCHITECTURE

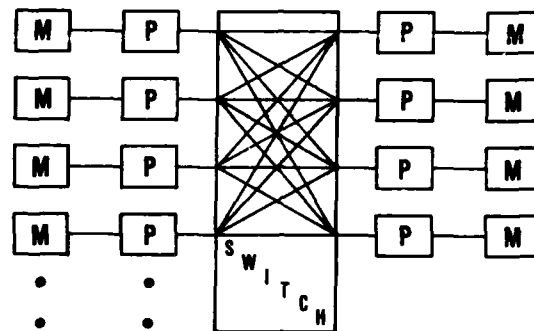
FIGURE 5



SI - SENSOR INPUT MODULE
 SP - SUPERVISORY PROCESSOR
 PN - DATA PROCESSORS
 Nn - NODE

GENERALIZED CIRCUIT - SWITCHED NETWORK TOPOLOGY

FIGURE 6

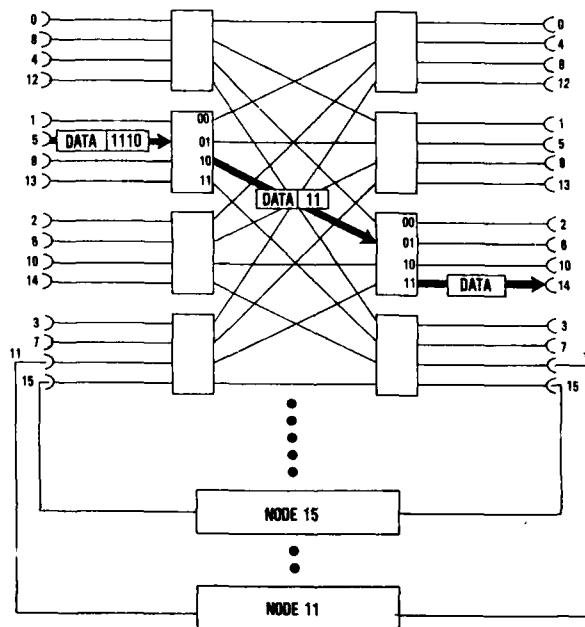


P~PROCESSOR

M~MEMORY

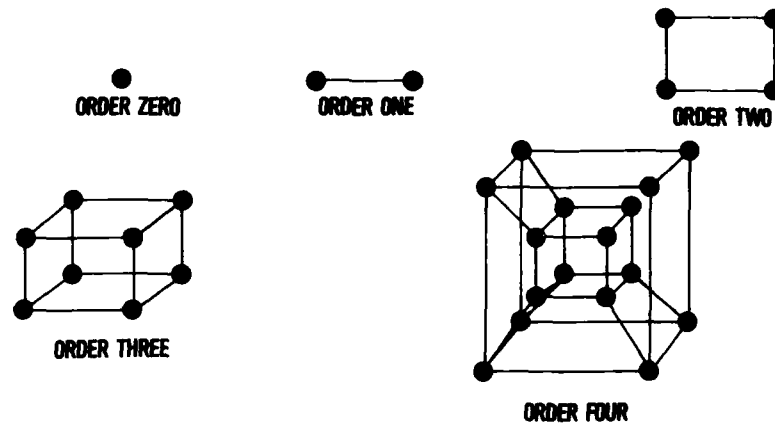
GENERALIZED SHARED MEMORY PACKET SWITCH

FIGURE 7



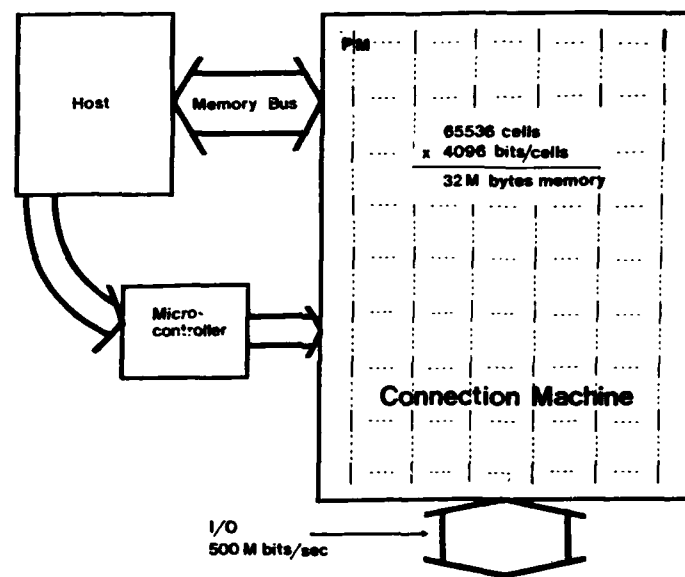
A PACKET IN TRANSIT THROUGH A BUTTERFLY SWITCH [5]

FIGURE 8



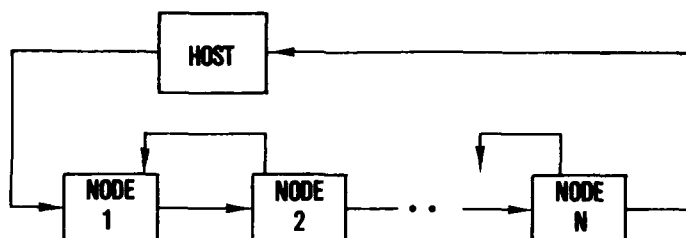
HYPERCUBE TOPOLOGY SHOWING VARIOUS ORDERS

FIGURE 9



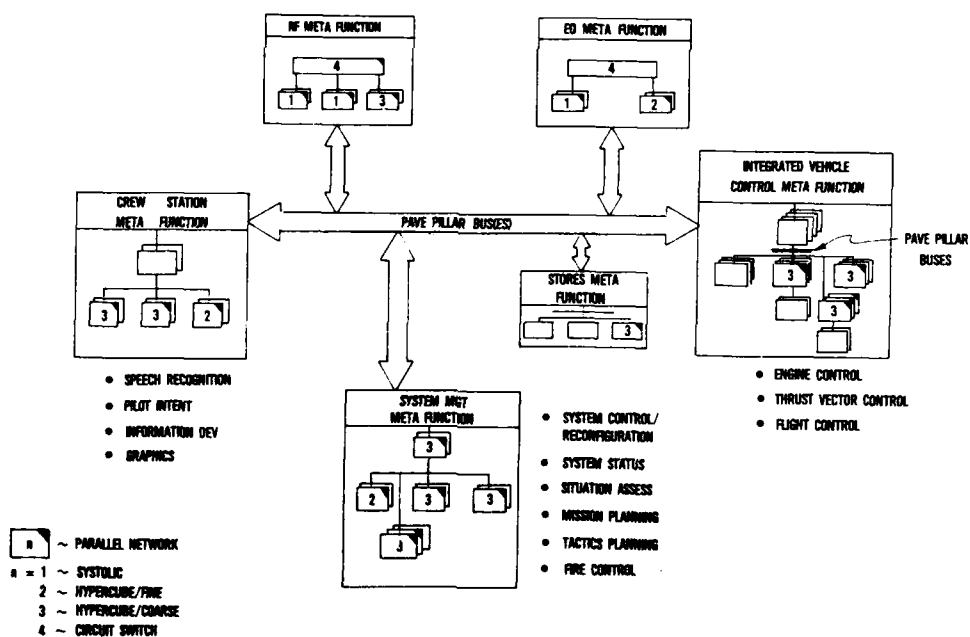
BLOCK DIAGRAM OF THE CM-1 PROTOTYPE CONNECTION MACHINE [1]

FIGURE 10



GENERALIZED SYSTOLIC ARRAY ARCHITECTURE

FIGURE 11



CONCEPTUAL ARCHITECTURE

FIGURE 12

SELECTIVE BIBLIOGRAPHY

This bibliography with abstracts has been prepared to support AGARD Lecture Series No.158 by the Centre de Documentation de l'Armement (CEDOCAR) of the French Ministry of Defence, in consultation with the Lecture Series Director, *Ingénieur Principal de l'Armement* L.Guibert of the Service Technique des Télécommunications et des Equipements Aéronautiques (STTE) of France.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-1- 229560 C.CEDOCAR

titre fr. : (L'impact des systemes informatiques avances sur la fiabilite de l'avionique militaire.)
titre ang. : The impact of advanced computer systems on avionics reliability.
Auteurs : BORDEN A. G.
type de doc. : Publication en serie
Titre publi. : Defense Electronics (US)
Source : VOL 19; NO 6; pp. 5 7-5 21; 12 Ref.; 6 Fig.; DP. 1987/05
CODEN : DEELDH
resume : Introduction, dans les systemes avioniques integrant toutes les fonctions d'aide au pilote, des techniques les plus avancees dans la science du traitement de l'information.

-2- 226254 C.CEDOCAR

titre fr. : Interconnexions de sous-systemes : bus de donnees.
titre ang. : Subsystems interconnection : data buses.
Titre conf. : IEEE/AIAA 7th digital avionics systems conferenc.
LIEU DE CONF. : Fort Worth (US)
DATE CONF. : 1986/10/13-1986/10/16
Auteur coll. : Institute of Electrical and Electronics Engineers et AIAA (US)
type de doc. : Memoire Congres
editeur : IEEE (new York)
Source : NO 86CH 23598; pp. 221-254; nb. Ref.; nb. Fig.; 5 communications; DP. 1986
resume : Description de divers bus de transmission de donnees developpes pour l'interconnexion des divers systemes avioniques a bord d'un aeronef.

-3- 225433 C.CEDOCAR

titre fr. : (Materiel de traitement de donnees pour spatonef : la norme 1750 A ISA de l'US Air Force constitue la nouvelle tendance.)
titre ang. : Data-processing hardware for spacecraft : Air Force standard 1750 A ISA is the new trend.
Auteurs : BYINGTON L.; THEIS D.
Affiliation : The Aerosp. Corp., Los Angeles, US; The Aerosp. Corp., Los Angeles, US
type de doc. : Publication en serie
Titre publi. : Computer (US)
Source : VOL 19; NO 11; pp. 50-59; 12 Ref.; 3 Fig.; 4 Tabl.; Contrat USAF No F04701-83-0084; DP. 1986
CODEN : CPTRB4
Issn : 0018-9162
resume : L'application de la norme 1750 A de l'US Air Force sur l'architecture des ensembles d'instructions (ISA) va dominer la technologie du traitement de donnees aerospaciales durant la prochaine decennie et va permettre d'embarquer des outils de developpement de logiciels plus sophistiques a bord des spatonefs. Presentation de l'etat actuel des materiels et logiciels homologues ou candidats pour des applications spatiales. La premiere famille (RCA 1802 series) de microprocesseurs spatiaux n'est desormais plus retenue.

-4- 221054 C.CEDOCAR

titre fr. : (Possibilite d'application aux avions futurs a hautes performances des developpements technologiques realises pour le F-4 et le Tornado.)
Titre all. : Uebertragbarkeit technologischer Weiterentwicklungen zu F-4 und Tornado auf zukünftige Hochleistungsflugzeuge.
Auteurs : KUNY W.
Affiliation : MBB (DE)
type de doc. : Publication en serie
Titre publi. : DGLR Jahrbuch (DE)
Source : VOL 1985/2; NO 41; 7 p.; 10 Fig.; DGLR 85 131; DP. 1986
CODEN : DGLJAN
Issn : 0070-4083
resume : Avantages des transferts de technologie et exemples

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

d'application au chasseur des années 90 dans les
domaines : architecture des systèmes, affichage,
logiciels, intégration des systèmes.

-5- 216791 C.CEDOCAR

titre fr. : Architecture de calculateurs tolérants les
défaillances.
titre ang. : Fault tolerant computer architecture.
Titre conf. : NAECON - IEEE 1986 National Aerospace and
Electronics Conference.
LIEU DE CONF. : Dayton, US
DATE CONF. : 1986/05/19-1986/05/23
Auteurs : BROCK L. D.; LALA J.; THOMPSON D. B.; BORTNER R. A.;
HUNG J. Y.
Affiliation : Charles Stark Draper Lab. (US); Charles Stark Draper
Lab. (US); Air Force Wright Aeronautical Labs (US);
Air Force Wright Aeronautical Labs (US); Air Force
Wright Aeronautical Labs (US)
Auteur coll. : Institute of Electrical and Electronics Engineers
(US)
type de doc. : Mémoire Congrès
éditeur : IEEE (New York)
Source : VOL 2; NO 86CH2307-7; pp. 368-382; 16 Ref.; 6 Fig.;
2 communications; DP. 1986
resume : Le système avance de traitement de données pour
véhicules aérospatiaux, conçu pour tolérer les
défaillances et les dommages, se détériore
progressivement, etc... Le CRMMFCS : le système de
commande de vol multi-microprocesseur a
reconfiguration continue.

-6- 216364 C.CEDOCAR

titre fr. : Fibres optiques.
titre ang. : Fiber optics.
Titre conf. : NAECON - IEEE 1986 National Aerospace and
Electronics Conference.
LIEU DE CONF. : Dayton, US
DATE CONF. : 1986/05/19-1986/05/23
Auteurs : MCLEAN J. R.; UHLHORN R. W.; FREEBERG D. M.; TOROK
E. J.; STANKOS W. C.
Affiliation : Harris Corp (US); Harris Corp (US); Sperry Corps
(US); Sperry Corps (US); Harris Corps (US)
Auteur coll. : Institute of Electrical and Electronics Engineers
(US)
type de doc. : Mémoire Congrès
éditeur : IEEE (New York)
Source : VOL 1; NO 86CH2307-7; pp. 146-164; Nb. Ref.; Nb.
Fig.; 3 communications; DP. 1986
resume : Réseau de transmission par fibres optiques à bord
d'un avion militaire : caractéristiques et
impératifs. Commutation optique pour bus de
transmission à grand débit. Optimisation d'un
système d'interconnexion démontable pour bus de
transmission de données par fibres optiques.

-7- 215570 C.CEDOCAR

titre fr. : Architectures d'avionique avancée.
titre ang. : Advanced avionics architecture.
Titre conf. : NAECON - IEEE 1985 National Aerospace and
Electronics Conference.
LIEU DE CONF. : Dayton, US
DATE CONF. : 1986/05/19-1986/05/23
Auteurs : HOEFER W. G.; ELENGICAL G.; MAKI S.; NORMAN L. E.; O
REILLY W. T.
Affiliation : General Electric (US); Westinghouse, Defense and
Electronics Syst. (US); General Dynamics (US); Texas
Instruments (US); Westinghouse (US)
Auteur coll. : Institute of Electrical and Electronics Engineers
(US)
type de doc. : Mémoire Congrès
éditeur : IEEE (New York)
Source : VOL 1; NO 86CH2307-7; pp. 112-143, 1386-1392; Nb.
Ref.; Nb. Fig.; 6 communications; DP. 1986
resume : Architecture d'un coprocesseur d'estimation de
paramètres. Evaluation de l'état de fonctionnement

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

des systemes d'armes aeroportes. Le calculateur correspondant a la norme MIL-STD-1750 A de Texas Instruments pour le traitement a bord des avions tactiques des donnees fournies par les differents capteurs. Une architecture redondante pour les differents capteurs. Une architecture redondante pour la tolerance aux defaillances des systemes d'armes aeroportes. Realisation de l'electronique des vehicules de transfert d'orbite utilisant les techniques de cout du cycle de vie. Des architectures developpees par Sanders multiplient par un facteur de deux a cinq le debit des processeurs MIL-STD-1750 A.

-8- 207440 C.CEDOCAR

titre fr. : Les outils des systemes experts et leurs applications.
titre ang. : Experts systems tools and applications.
Titre conf. : Conference on software tools.
LIEU DE CONF. : New York, US
DATE CONF. : 1985/04/14-1985/04/17
Auteur coll. : Polytech. Inst. of New York (US)
type de doc. : Memoire Congres
editeur : IEEE Computer Society Press (Washington)
Source : NO 85CH2188-1; pp. 28-70; 38 Ref.; 19 Fig.; 14 Phot.; 6 memoires; DP. 1985
Isbn : 0-818-60628-2
resume : Description : du programme d'edition DBIE-1 developpe pour des simulations interactives (notamment de tableau de bord d'aeronef pour l'entrainement des equipages); du logiciel LANST concu pour la simulation de reseau locaux; des systemes experts BES/SSD d'aide a la mise au point et a la correction de logiciels de grandes dimensions; de l'outil SPACE II d'aide en developpement et a la maintenance des logiciels (generalement en langage ADA) des forces armees americaines; d'un systeme expert d'aide a la formulation de la politique monetaire des Etats-Unis; de l'outil de peinture graphique DA VINCI avec base de connaissance.

-9- 203927 C.CEDOCAR

titre fr. : Developpement d'un logiciel de gestion d'interfaces systeme.
Auteurs : MOREL J. H.; MOURRE B.
Auteur coll. : Ec. Natl. Super. d'Ing. de Constr. Aeronaut. (FR)
type de doc. : Rapport
no brevet : ENSICA 85/6
Source : Rapport de fin d'etudes; 71 p.; 5 Ref.; NB Fig.; NB Tabl.; DP. 1985
resume : Le digibus. Les coupleurs d'accès bus. Presentation du logiciel. Programme d'acquisition des parametres. Programme de test des coherences.

-10- 199718 C.CEDOCAR

titre fr. : Architecture de calculateur numerique pour un systeme de commande de vol de conception avancee.
titre ang. : Digital computer architecture as applied to an advanced flight control system.
Titre conf. : AIAA Guidance, Navigation and Control Conference.
LIEU DE CONF. : Snowmass, US
DATE CONF. : 1985/08/19-1985/08/21
Auteurs : DIETRICH A. R.; THOMAS F. J.
Affiliation : Allied Bendix Aerospace (US); Allied Bendix Aerospace (US)
Auteur coll. : AIAA (US)
type de doc. : Memoire Congres
editeur : The American Institute of Aeronautics and Astronautics, New York
Source : pp. 836-841; 3 Fig.; 1 Tabl.; DP. 1985
resume : Etude de l'architecture de ce calculateur et de la redondance requise pour assurer la survie de l'aeronef en cas d'une defaillance de circuit ou d'une erreur de logiciel. Ce systeme de commande

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

utilise des circuits integres a tres grande echelle
(VLSI) avec des bus de transmission numerique tres
rapides.

-11- 183235 C.CEDOCAR

titre fr. : Bus de transmission de donnees.
titre ang. : Data bus concepts and practices.
Titre conf. : AIAA/IEEE 6th digital avionics systems conference.
LIEU DE CONF. : Baltimore (US)
DATE CONF. : 1984/12/03-1984/12/06
Auteurs : PENN D.; CHOW K. K.; SPIETH J.; MACNAMERA B. J.;
DOLECEK C. E.
Affiliation : Israel Aircraft Ind (IL); Lockheed (US); US Air
Force (US); Arinc (US); John Hopkins Univ. (US)
Auteur coll. : American Institute of Aeronautics and Astronautics
(US)
type de doc. : Memoire Congres
editeur : AIAA, New York
Source : pp. 393-420; 22 Ref.; 35 Fig.; 1 Tabl.; 5
communications; DP. 1984
resume : Un bus standard pour les calculateurs embarques
MIL-STD-1750A. Un bus de transmission de donnees
optique par multiplexage de longueur d'onde pour
applications militaires. L'interface MIL-STD-1553 B
ARINC 561. Utilisation de l'intelligence
artificielle dans les reseaux locaux CSMA/CD et a
jetons. Structures de bus tres rapides pour le
multitraitement.

-12- 182685 C.CEDOCAR

titre fr. : Circuits integres numeriques avances.
titre ang. : Advanced digital integrated circuits.
Titre conf. : AIAA/IEEE 6th digital avionics systems conference.
LIEU DE CONF. : Baltimore (US)
DATE CONF. : 1984/12/03-1984/12/06
Auteurs : FRIEDMAN S. N.; FORDE S. J.; HILMANTEL M. A.
Affiliation : ILC data service corp (US); Sanders associate (US);
Sanders associate (US)
Auteur coll. : American Institute of Aeronautics and Astronautics
(US)
type de doc. : Memoire Congres
editeur : AIAA, New York
Source : pp. 563-572; 1 Ref.; 9 Fig.; 2 communications; DP.
1984
resume : Performances, description et caracteristiques
electriques du BUS 65112 suprahhybride, unite pour
terminal a distance, conforme a la norme
MIL-STD-1553B. Circuit VLSI pour calculateur
embarque.

-13- 182675 C.CEDOCAR

titre fr. : Le calculateur aeroporte de l'avenir.
Auteurs : CONDOM P.; BULLOCH C.
type de doc. : Publication en serie
Titre publi. : Interavia Revue (CH)
Source : NO 1; pp. 50-53; 3 Fig.; DP. 1985/01
CODEN : INRVB0
Issn : 0376-9402
resume : Ensemble de deux articles relatifs d'une part aux
libertes et aux contraintes des commandes de vol
electriques et d'autre part au calculateur aeroporte
de l'avenir qui deviendra de plus en plus compact,
rapide et intelligent.

-14- 182229 C.CEDOCAR

titre fr. : Avionique modulaire standard.
titre ang. : Standardized modular avionics.
Titre conf. : AIAA/IEEE 6th digital avionics systems conference.
LIEU DE CONF. : Baltimore (US)
DATE CONF. : 1984/12/03-1984/12/06
Auteurs : MULVANEY S. P.; PORADISH F. J.; MCNICHOLS J. K.;
RADCLIFFE W. E.; KUSEK J. L.
Affiliation : General Dynamics (US); Texas Instruments (US); Naval
Avionics Center (US); ARINC (US); General Dynamics
(US)

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

- Auteur coll. : American Institute of Aeronautics and Astronautics (US)
type de doc. : Memoire Congres
editeur : AIAA, New York
Source : pp. 624-652; 5 Ref.; 34 Fig.; 3 Tabl.; 5 communications; DP. 1984
resume : L'avionique de construction modulaire. Un calculateur entierement reconfigurable.
- 15- 178847 C.CEDOCAR
titre fr. : Developpement d'une famille de calculateurs modulaires pour le traitement de donnees de vol a cout d'acquisition minimal.
titre ang. : Development of a modular air data computer family for minimum cost of ownership.
Titre conf. : 14th congress of ICAS.
LIEU DE CONF. : Toulouse (FR)
DATE CONF. : 1984/09/10-1984/09/14
Auteurs : COLSON J. M.; MACKLEY F. T.
Affiliation : Marconi Avionics Ltd, Rochester (GB); Marconi Avionics Ltd, Rochester (GB)
type de doc. : Memoire Congres
Titre publi. : ICAS Congress (US)
editeur : ICAS
Source : VOL 2; NO 4-9-3; pp. 1075-1078; 1 Fig.; 4 Phot.; DP. 1984
CODEN : AAUSAK
Issn : 0-915-92889-2
resume : Les developpements de l'utilisation de la conception assistee par calculateur et de l'integration a tres grande echelle ont permis de concevoir et fabriquer quatre variantes de calculateurs numeriques, possedant un tronc commun electronique de 85 pour cent et pouvant etre adaptees a 30 types distincts d'aeronefs. Etude des parametres economiques de ces calculateurs. Description du hardware et des essais en vol.
- 16- 174497 C.CEDOCAR
titre fr. : (Architecture des calculateurs destines a l'avionique.)
titre ang. : Avionics computer architecture.
Auteurs : NICHILS J.
type de doc. : Publication en serie
Titre publi. : International Defense Review (CH)
Source : VOL 17/9; pp. 50-53; 8 Fig.; Special Electronics No 2/84; DP. 1984
CODEN : IDRVAL
Issn : 0020-6512
resume : Diverses categories d'architectures de calculateurs. Les calculateurs sequentiels. Exemples de multiprocesseurs pour systemes avioniques. Les premiers processeurs paralleles aeroportes.
- 17- 163931 C.CEDOCAR
titre fr. : L'avionique du F/A-18.
titre ang. : Advanced F/A-18 avionics.
Titre conf. : Advanced concepts for avionics/weapon system design, development and integration.
LIEU DE CONF. : Ottawa (Canada)
DATE CONF. : 1983/04/18-1983/04/22
Auteurs : DRUMMOND R. C.; LOOPER J. L.
Affiliation : McDonnell Douglas; McDonnell Douglas
Auteur coll. : Advisory group for aerospace reserch and development
type de doc. : Memoire Congres
Titre publi. : Agard conference proceedings (FR)
editeur : Agard (Neuilly/Seine)
Source : NO CP 343; pp. 15.1-15.12; 12 Fig.; DP. 1983
CODEN : AGCPAV
Issn : 0549-7191
resume : Description du systeme avionique/systeme d'arme. Circuits integres a grande echelle et microprocesseurs d'apres la norme MIL STD 1553A. Missile AMRAAM. Systeme d'information JTIDS.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-18- 162950 C.CEDOCAR

titre fr. : Conference sur les calculateurs dans le domaine
aerospacial.
Titre conf. : Computers in Aerospace Conference.
LIEU DE CONF. : Hartford, CT, Etats-Unis
DATE CONF. : 1983/10/24-1983/10/26
Auteur coll. : Amer. Inst. of Aeron. and Astronautics
type de doc. : Congres
editeur : Amer. Inst. of Aeron. and Astronautics
Source : NO 4; 502 p.; DP. 1983
resume : Recueil de 68 memoires signales en detail dans les
International Aerospace Abstracts volume 24, No 1 du
01/01/84.

-19- 152566 C.CEDOCAR

titre fr. : (Sur les systemes numeriques).
titre ang. : Digital system topics
Auteurs : CAMPBELL H. W.
Auteur coll. : Institute of Electrical and Electronics Engineers
type de doc. : Publication en Serie
Titre publi. : IEEE Conference Publication (US)
Source : VOL. 1, NO 83CH1868-9 (1983), PP. 670-694; 17 ref.,
17 fig., 4 tabl. NAECON 83, Dayton, OH, 17-19/05/83
CODEN : ICHQ24
resume : Quatre articles traitent: d'un programme totalement
interactif pour l'evaluation de la fiabilite de
systemes numeriques tolerant les erreurs; de
techniques d'analyse de fiabilite et de soutien
logistique pour avionique tolerant les erreurs; de
simulations appliquees a la conception d'un systeme
multiplex integre pour l'avionique pour helicoptere
Agusta A 129; d'une architecture pour systemes
d'informatique repartie fonctionnant en temps reel.

-20- 144755 C.CEDOCAR

titre fr. : (L'armee de l'air americaine etudie un nouveau
concept de calculateur).
titre ang. : USAF studies new computer concept
Auteurs : ELSON B. M.
type de doc. : Publication en Serie
Titre publi. : Aviation Week and Space Technology (US)
Source : VOL. 118, NO 19 (9/5/83), PP. 69-71; 2 fig.
CODEN : AWSTAV
resume : Le laboratoire de dynamique de vol de l'US Air Force
evalue une architecture de calculateur
multimicroprocesseur pour commande de vol
susceptible de reduire largement le cout des
calculateurs tolerant les defaillances et de
multiplier plusieurs fois le temps disponible entre
les operations de maintenance planifiee de ces
systemes. Des microprocesseurs autonomes sont
reliees par un ensemble commun de bus de donnees.
Ils comprennent des unites actives et des unites en
reserve (60 et 40 environ respectivement), chaque
processeur etant capable d'effectuer n'importe
quelle tache a n'importe quel moment.

-21- 144126 C.CEDOCAR

titre fr. : (Microcalculateur operationnel tolerant les
defaillances pour une fiabilite tres elevee).
titre ang. : Operational fault-tolerant microcomputer for very
high reliability
Auteurs : MASHKURI YAACOB; HARTLEY M. G.; DEPLEDGE P. G.
Affiliation : (1) Univ. of Malaya, Malaisie, (2,3) Univ. of
Manchester (GB)
type de doc. : Publication en Serie
Titre publi. : IEE Proceedings-E-Computers and Digital Technique
(US)
Source : VOL. 130, NO 3 (5/83), PP. 90-94; 17 ref., 4 fig.,
1 tabl.
CODEN : IJCTDJ
Issn : 0143-7062
resume : Description d'un microcalculateur a triple
redondance modulaire concu pour fonctionner pendant
3 heures a bord d'un aeronef, avec une probabilite

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

de panne inferieure a 1 sur 10 exp. B. Il utilise un
reseau de microprocesseurs Motorola M 6800.

-22- 139562 C.CEDOCAR

titre fr. : (Systemes de gestion de vol et systemes de
communication).
titre ang. : Flight management systems and data links
Auteurs : HENDRICKSON T. W.
Affiliation : (1) BOEING COMPANY
type de doc. : Publication en Serie
Titre publi. : The aeronautical journal (GB)
Source : VOL. 87, NO 862 (02/83), PP. 52-67; 6 ref. , 20
fig., 5 tabl. No 1054. Conference -Today and tomorrow
mini and micro computers in airline operations.
10/82
CODEN : AENJAK
Issn : 0001-9240
resume : Presentation de l'architecture et des interfaces du
systeme digital de gestion de vol mis au point par
Boeing pour le 757 et 767.

-23- 131699 C.CEDOCAR

titre fr. : (Application de l'informatique distribuee aux
systemes avioniques).
titre ang. : Application of distributed system designs to avionic
systems
Auteurs : MOSES K.; NELSON H.; WILCOCK G. W.; LANCASTER P.
Affiliation : (1) Bendix Corp (US), (2) Navi Weapons Center (US),
(3) Royal Aircraft Establishment (GB), (4) British
Aerospace (GB)
type de doc. : Publication en Serie
Titre publi. : AGARD Conference Proceedings
Source : VOL. CP-303 (1981), PP. 32.1-57.3; nbr. ref. , nbr.
fig., nbr. tabl. Tactical Airborne Distributed
Computing and Networks, Roros 22-25/6/81
CODEN : AGCPAV
resume : Sept communications. Commande integree des systemes
mecaniques pour le futur avion de combat.
Architecture du systeme d'armes du Mirage 2000.
Calculateurs et unites de soutien du systeme d'armes
des avions F/A-18 et F/A-18A. Conception d'un reseau
informatique ultra-fiable pour avionique.

-24- 131697 C.CEDOCAR

titre fr. : (Tolerance aux defaillances et fiabilite dans la
conception).
titre ang. : Fault tolerance and reliability in designs
Auteurs : STERN A. D.; MCGOUGH J. G.; SWERN F.; BAVUSO S. J.
Affiliation : (1) Boeing Co. (US), (2,3) Bendix Corp. (US), (4)
Nasa (US)
type de doc. : Publication en Serie
Titre publi. : AGARD Conference Proceedings
Source : VOL. CP-303 (1981), PP. 20.1-55.2; nbr. ref. , nbr.
fig., nbr. tabl. Tactical Airborne Distributed
Computing and Networks Roros 22-25/6/81
CODEN : AGCPAV
resume : Cinq communications. Methodologie pour evaluer le
risque de panne d'un miniprocesseur avionique
numerique. Specification hierarchisee d'un systeme
de commande de vol tolerant des defaillances. Reseau
d'echange reconfigurable pour controle de processus
reparti.

-25- 130669 C.CEDOCAR

titre fr. : (Methodes de conception d'un systeme distribue).
titre ang. : Distributed system design approaches
Auteurs : CALLAWAY A. A.; PRIESTER R. W.; BROMLEY K.; CLARY J.
Affiliation : (1) Royal Aircraft Establishment (GB), (2) Research
Triangle Inst. (US), (3,4) Naval Ocean Systems
Center (US)
type de doc. : Publication en Serie
Titre publi. : AGARD Conference Proceedings
Source : VOL. CP-303 (1981), PP. 11.1-53.3; nbr. ref. , nbr.
fig., nbr. tabl. Tactical Airborne Distributed
Computing and Networks, Roros 22-25/6/81)

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

- CODEN : AGCPAV
Isbn : 92-835-0302-3
resume : Quatre communications. Presentation d'une technique de gestion de base de donnees destinee a verifier l'architecture d'un systeme d'informatique repartie. Methodes de traitement de signal a l'aide de reseaux systoliques. Commande informatisee distribuee en temps reel pour les systemes avionique des aeronefs embarques sur porte avions.
- 26- 129190 C.CEDOCAR
titre fr. : Les calculateurs de gestion de vol.
Auteurs : GROSSIN M. J.
type de doc. : Ouvrage
editeur : Institut Francais de Navigation
Source : VOL. 1, NO FP 4 (1982), 17 p.; 10 fig. Congres International des Instituts de Navigation, Paris, 21-24/09/82
resume : Equipement des avions civils de la nouvelle generation avec un systeme FMS unique pour la navigation horizontale et verticale: architecture du FMS et optimisation du vol.
- 27- 125245 C.CEDOCAR
titre fr. : (Specification formelle et verification mecanique du calculateur SIFT: un systeme de commande de vol tolerant les defaillances).
titre ang. : Formal specification and mechanical verification of SIFT: a fault-tolerant flight control system
Auteurs : MELLIAR SMITH P. M.; SCHWARTZ R. L.
Affiliation : (1,2) SRI Intern., US
type de doc. : Publication en Serie
Titre publi. : IEEE Transactions on Computers (US)
Source : VOL. 31, NO 7 (07/82), PP. 616-630; 11 ref. . 9 fig.
CODEN : ITCOB4
Issn : 0018-9340
resume : Presentation des methodes d'essai employees pour demontrer que ce calculateur repondait aux specifications de tolerance de defaillance propres aux systemes de commande de vol, telles qu'elles ont ete fixees par la FAA et la NASA.
- 28- 107681 C.CEDOCAR
titre fr. : (Le LHX: une conception de systeme avionique avance).
titre ang. : LHX: an advanced avionics system design
Auteurs : D AVINO D. S.; SPIEGEL S. S.
type de doc. : Publication en Serie
Titre publi. : AIAA/IEEE Digital Avionics Conference (US)
Source : VOL. 4, NO 2249 (11/81), PP. 156-162; 5 fig. (St Louis (Mo.), 17-19/11/81)
CODEN : AAPRAQ
resume : Description de l'architecture de cet equipement avionique destine aux systemes d'armes avancees des annees 90 et en particulier aux helicopteres.
- 29- 102277 C.CEDOCAR
titre fr. : Software : design and control of software dominated systems; (Le logiciel : conception et commande des systemes ou le logiciel joue un role dominant).
Auteurs : CACERES R. G.; WARD J. W.; BERLACK H. R.; TOULOUSE P.; SUTTON R. W.
Auteur coll. : Radio Technical Commission for Aeronautics
type de doc. : Publication en Serie
Titre publi. : Proceedings of the 1981 RTCA Assembly (US)
no brevet : SBN 0145-9589
Source : NO 1981 (1981), PP. 123-213; nombr. fig. (3e Session)
CODEN : RTCABR
resume : Cinq memoires : conception du logiciel pour la systeme de guidage numerique du DC-9 Super 80; application des normes de conception du logiciel a l'equipement des avions commerciaux; controle de la configuration du materiel et du logiciel; le logiciel dans l'homologation des nouveaux avions

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

civils de transport europeens; le logiciel dans
l'homologation des systemes et equipements de bord.

-30- 74757 C.CEDOCAR

titre ang. : Software standardization and MIL-STD-1750.
Auteurs : HERRELKO D. A.; DENTON D.
type de doc. : Publication en Serie
Titre publi. : Institute of Electrical and Electronics Engineers,
New York, National Aerospace and electronics
conference
Source : VOL. 2, NO 1980 (1980), PP. 880-91; 25 ref. (Dayton
(Ohio), 20-22/5/80, A81-30226)
CODEN : ICH024

-31- 74736 C.CEDOCAR

titre ang. : Fault tolerance applications to future military
system avionics.
Auteurs : HARRIS R. L.; JONES E. E.
type de doc. : Publication en Serie
Titre publi. : Institute of Electrical and Electronics Engineers,
New York, National Aerospace and electronics
conference
Source : VOL. 1, NO 1980 (1980), PP. 412-419; 20 ref. (Dayton
(Ohio), 20-22/5/80, A81-30226)
CODEN : ICH024

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-1- 618419 C.NTIS

Titre anglais : New Technology Impacts on Future Avionics Architectures.

Ref. Conference : This article is from 'Advanced Computer Aids in the Planning and Execution of Air Warfare and Ground Strike Operations: Conference Proceedings, Meeting of the Avionics Panels of AGARD (51st) Held in Kongsberg, Norway on 12-16 May 1986,' AD-A182 096, p10-1-10-7.

Auteurs : MEJZAK R. S.

Organisme auteur : Naval Air Development Center, Warminster, PA.

Type Document : Conference

Source : NP. 7; NTIS Prices: PC A02/MF A01; DP. c1986

Resume : This paper provides an interpretation of avionics architecture with respect to system components, organization, and design factors. Initially, general avionics architecture characteristics are addressed followed by discussions on emerging new technologies and their impact and advanced systems. Information handling requirements are projected for future tactical aircraft. In addition, advanced avionics architecture design consideration and technical issues are addressed relative to achieving improved performance, reliability, survivability, flexibility and low life cycle cost. (Author).

-2- 616504 C.NTIS

Titre anglais : Impact of Future Avionics Technology on the Conduct of Air Warfare.

Auteurs : DASARO J. A.

Organisme auteur : Army Avionics Research and Development Activity, Fort Monmouth, NJ.*National Aeronautics and Space Administration, Washington, DC.

Type Document : Report

Source : In AGARD Improvement of Combat Performance for Existing and Future Aircraft, 9p.; NP. 9; NTIS Prices: (Order as N87-22663 PC A07/MF A01); DP. Dec 86

Resume : A synopsis is given of the conclusions reached by the Systems Subpanel of the NATO AGARD workshop on the potential impact of development in electronic technology on the future conduct of air warfare. Avionic system integration technology, system architecture, data processing, data communication paths, computer programs, fault detection and isolation, and system design methodology are among the topics discussed.

-3- 602192 C.NTIS

Titre anglais : Potential Impact of Developments in Electronic Technology on the Future Conduct of Air Warfare. Volume 3.

Organisme auteur : Advisory Group for Aerospace Research and Development, Neuilly-sur-Seine (France).

Type Document : Report

RPT,CTR,GRT : RPT AGARD-AR-232-VOL-3

Source : See also Volume 2, AD-CO40 122. Presented at AGARD Avionics Panel Workshop, The Hague, Netherlands, 21-25 Oct 85.; NP. 118; NTIS Prices: PC A06/MF A01; DP. Dec 86

Resume : Today the Warsaw Pact nations enjoy a large quantitative advantage in equipment that is offset by superior technology in our weapons systems. Nevertheless, most agree that continued exploitation of emerging technologies is needed by NATO to maintain a qualitative lead, and that continued progress in microelectronics and information processing offers opportunities for NATO to maintain its technical superiority. The development of electronic technology continues unabated. Advances in microelectronics are resulting in circuit densities many orders of magnitude greater than in current usage, making possible higher speed circuitry and greater storage capacity. At the same time, evolving RG techniques are leading to

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

monolithic microwave integrated circuits and microstrip antennas with corresponding reductions in size and weight. In addition, rapid advances are taking place in computer architecture and software that will provide improved information processing and control. This, coupled with progress in artificial intelligence and man-machine interface, offers promise of greatly improved battle management in the cockpit. The scale of changes is such that the nature of air warfare should be significantly affected over the next twenty years. (NATO furnished).

-4- 595437 C.NTIS

Titre anglais : Aircraft Integrated Data System (AIDS).
Ref. Conference : 13. symposium on Aircraft Integrated Data Systems (AIDS-13), Hamburg (Germany, F.R.), 17-19 Sep 1985.
Auteurs : KALBE H.
Organisme auteur : Messerschmitt-Boelkow-Blohm G.m.b.H., Munich (Germany, F.R.). Information und Dokumentation.
Type Document : Conference
RPT,CTR,GRT : RPT MBB-UT--67/85-0e
Source : NP. 13; NTIS Prices: PC E07; DP. 1985
Resume : The experiences with the Aircraft Integrated Data System (AIDS) over several years in the A-310 are reported. The situation referring to the customer requirements, the systems goals and the integration problems are discussed. The repercussions of these aspects on the systems architecture and a solution which covers these problems are presented. (RMM). (Copyright (c) 1986 by FIZ. Citation no. 86:080824.).

-5- 585327 C.NTIS

Titre anglais : Synergistically Integrated Reliability Architecture: A Reliability Analysis of an Ultra-Reliable Fault Tolerant Computer Design.
Auteurs : NELSON D. J.
Organisme auteur : Naval Postgraduate School, Monterey, CA.
Type Document : Thesis
Source : Master's thesis; NP. 137; NTIS Prices: PC A07/MF A01; DP. 26 Sep 86
Resume : This thesis develops a Semi-Markov reliability model for the Synergistically Integrated Reliability (SIR) computer architecture. The SIR architecture is an advanced hybrid redundancy scheme that combines several current reliability techniques to achieve hardware and software reliability. These methods include hybrid redundancy, N-Version programming and source congruent data interchange. The architecture is designed to support active control systems in the aircraft industry as well as the bus controller requirements for the Dispersed Sensor Processor Mesh (DSPM) system for ultra-reliable computer communications. The paper also develops high level algorithms for fault detection, location, and configuration management within the SIR system. The reliability model integrates the hardware design, the hybrid redundancy philosophy, and the operating constraints of an active control system into a single reliability model. Specific models are developed for the 3, 4, and 5 processor cases of the SIR architecture and plots of the system reliability vs mission time are generated using the SURE Reliability Analysis Program.

-6- 566524 C.NTIS

Titre anglais : Investigation of an Advanced Fault Tolerant Integrated Avionics System.
Auteurs : DUNN W. R.; COTTRELL D.; FLANDERS J.; JAVORNIK A.; RUSOVICK M.
Organisme auteur : University of Southern Colorado, Pueblo. School of Applied Science and Engineering Technology.*National Aeronautics and Space Administration, Washington, DC.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Type Document : Report
RPT,CTR,GRT : RPT NAS 1.26:176980; CTR NCC2-277
Source : Final technical rept. Nov 83-Mar 86.; NP. 76; NTIS
Prices: PC A05/MF A01; Monitor NASA-CR-176980; DP.
Mar 86

Resume : Presented is an advanced, fault-tolerant
multiprocessor avionics architecture as could be
employed in an advanced rotorcraft such as LHX. The
processor structure is designed to interface with
existing digital avionics systems and concepts
including the Army Digital Avionics System (ADAS)
cockpit/display system, navaid and communications
suites, integrated sensing suite, and the Advanced
Digital Optical Control System (ADOCS). The report
defines mission, maintenance and safety-of-flight
reliability goals as might be expected for an
operational LHX aircraft. Based on use of a modular,
compact (16-bit) microprocessor card family, results
of a preliminary study examining simplex, dual and
standby-sparing architectures is presented. Given
the stated constraints, it is shown that the dual
architecture is best suited to meet reliability
goals with minimum hardware and software overhead.
The report presents hardware and software design
considerations for realizing the architecture
including redundancy management requirements and
techniques as well as verification and validation
needs and methods.

-7- 532889 C.NTIS
Titre anglais : Reliability Aspects of Software for Digital
Avionics.
Auteurs : DEKKER G. J.
Organisme auteur : National Aerospace Lab., Amsterdam
(Netherlands). *National Aeronautics and Space
Administration, Washington, DC.

Type Document : Report
RPT,CTR,GRT : RPT NLR-TR-82126-U; RPT B8568394; CTR RB-RLB-1982-1-
3.3
Source : Revised.; NP. 95; NTIS Prices: PC A05/MF A01; DP.
1985

Resume : Methods to develop reliable software based avionics
systems, especially for safety critical functions,
are reviewed. The differences between analog and
digital systems, and the policy of the FAA to
certify software based systems are presented.
Methods to minimize the number of errors during
software development, methods to remove as many
errors as possible via testing, and methods to
minimize the effect of remaining errors during
operational flights are outlined. A safety analysis
regarding common-mode failures is given. Reliability
related techniques used by avionics manufacturers
are discussed.

-8- 498229 C.NTIS
Titre anglais : Development and Evaluation of a Fault-Tolerant
Multiprocessor (FTMP) Computer. Volume 3. FTMP Test
and Evaluation.
Auteurs : LALA J. H.; SMITH T. B.
Organisme auteur : Charles Stark Draper Lab., Inc., Cambridge,
MA. *National Aeronautics and Space Administration,
Washington, DC.

Type Document : Report
RPT,CTR,GRT : RPT NAS 1.26:166073; RPT CSDL-R-1602-V-3; CTR NAS1-
15336
Source : Final rept.; NP. 115; NTIS Prices: PC A06/MF A01;
Monitor NASA-CR-166073; DP. May 83

Resume : The experimental test and evaluation of the
Fault-Tolerant Multiprocessor (FTMP) is described.
Major objectives of this exercise include expanding
validation envelope, building confidence in the
system, revealing any weaknesses in the
architectural concepts and in their execution in
hardware and software, and in general, stressing the

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

hardware and software. To this end, pin-level faults were injected into one LRU of the FTMP and the FTMP response was measured in terms of fault detection, isolation, and recovery times. A total of 21,055 stuck-at-0, stuck-at-1 and invert-signal faults were injected in the CPU, memory, bus interface circuits, Bus Guardian Units, and voters and error latches. Of these, 17,418 were detected. At least 80 percent of undetected faults are estimated to be on unused pins. The multiprocessor identified all detected faults correctly and recovered successfully in each case. Total recovery time for all faults averaged a little over one second. This can be reduced to half a second by including appropriate self-tests.

-9- 498226 C.NTIS

Titre anglais : Development and Evaluation of a Fault-Tolerant Multiprocessor (FTMP) Computer. Volume 2. FTMP Software.

Auteurs : LALA J. H.; SMITH T. B.

Organisme auteur : Charles Stark Draper Lab., Inc., Cambridge, MA.*National Aeronautics and Space Administration, Washington, DC.

Type Document : Report

RPT,CTR,GRT : RPT NAS 1.26:166072; RPT CSDL-R-1601-V-2; CTR NAS1-15336

Source : Final rept.; NP. 234; NTIS Prices: PC A11/MF A01; Monitor NASA-CR-166072; DP. May 83

Resume : The software developed for the Fault-Tolerant Multiprocessor (FTMP) is described. The FTMP executive is a timer-interrupt driven dispatcher that schedules iterative tasks which run at 3.125, 12.5, and 25 Hz. Major tasks which run under the executive include system configuration control, flight control, and display. The flight control task includes autopilot and autoland functions for a jet transport aircraft. System Displays include status displays of all hardware elements (processors, memories, I/O ports, buses), failure log displays showing transient and hard faults, and an autopilot display. All software is in a higher order language (AED, an ALGOL derivative). The executive is a fully distributed general purpose executive which automatically balances the load among available processor triads. Provisions for graceful performance degradation under processing overload are an integral part of the scheduling algorithms.

-10- 496026 C.NTIS

Titre anglais : Navy Should Join the Air Force and Army Program to Develop an Advanced Integrated Avionics System.

Organisme auteur : General Accounting Office, Washington, DC. National Security and International Affairs Div.

Type Document : Report

RPT,CTR,GRT : RPT GAO/NSIAD-85-94; RPT B-215379

Source : NP. 17; NTIS Prices: PC A02/MF A01; DP. 17 Jun 85

Resume : Modern technology should soon enable separate avionics systems in an aircraft to be consolidated into a single package to conserve space, save weight, and reduce costs. The report points out the potential benefits of avionics consolidation and recommends the Navy join in a demonstration program now being conducted by the Air Force and Army to exploit such benefits.

-11- 487455 C.NTIS

Titre anglais : Rotorcraft Digital Advanced Avionics System (RDAAS) Functional Description.

Auteurs : PETERSON E. M.; BAILEY J.; MCMAHUS T. J.

Organisme auteur : Honeywell, Inc., St. Louis Park, MN. Military Avionics Div.*National Aeronautics and Space Administration, Washington, DC.

Type Document : Report

RPT,CTR,GRT : RPT NAS 1.26:166611; CTR NAS2-11665

Source : NP. 188; NTIS Prices: PC A08/MF A01; Monitor

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Resume : NASA-CR-166611; DP. Jan 85
: A functional design of a rotorcraft digital advanced avionics system (RODAAS) to transfer the technology developed for general aviation in the Demonstration Advanced Avionics System (DAAS) program to rotorcraft operation was undertaken. The objective was to develop an integrated avionics system design that enhances rotorcraft single pilot IFR operations without increasing the required pilot training/experience by exploiting advanced technology in computers, busing, displays and integrated systems design. A key element of the avionics system is the functionally distributed architecture that has the potential for high reliability with low weight, power and cost. A functional description of the RODAAS hardware and software functions is presented.

-12- 466401 C.NTIS

Titre anglais : Proceedings of the Technical Forum (3rd) on the F-16 MIL-STD-1750A Microprocessor and the F-16 MIL-STD-1589B Compiler Held at Wright-Patterson AFB, OH on May 5-6, 1982. Volume 2. Specifications.
Auteurs : PESLER J. L.; SHOULDERS D.
Organisme auteur : Aeronautical Systems Div., Wright-Patterson AFB, OH.
Type Document : Report
RPT,CTR,GRT : RPT ASD-TR-82-5011-VOL-2
Source : See also Volume 1, AD-A150 583.; NP. 547; NTIS Prices: PC A23/MF A01; DP. 6 May 82
Resume : Contents: Critical Item Development Specification for MIL-STD-1750A Microprocessors 16ZE181 (Preliminary); Avionic Processor Standard Instruction Set Architecture Requirements 16PP379A; Interim Processor Design Requirements 16PP456 (Draft); Computer Program Development Specification for the F-16 Integrated JOVIAL J-73 Support Software System 16ZE165; Dual MIL-STD-1750A Assembly Syntax for F-16A & Integrated Support Software System; Dual Users Manual.

-13- 466400 C.NTIS

Titre anglais : Proceedings of the Technical Forum (3rd) on the F-16 MIL-STD-1750A Microprocessor and the F-16 MIL-STD-1589B Compiler Held at Wright-Patterson AFB, OH on May 5-6, 1982. Volume 1. Papers.
Auteurs : PESLER J. L.; SHOULDERS D.
Organisme auteur : Aeronautical Systems Div., Wright-Patterson AFB, OH.
Type Document : Report
RPT,CTR,GRT : RPT ASD-TR-82-5011-VOL-1
Source : See also Volume 2, AD-A150 584.; NP. 547; NTIS Prices: PC A23/MF A01; DP. 6 May 82
Resume : Contents: MIL-STD-1750A Microprocessor: Management Overview; Program Overview; Functional and technical description; Applications and hardware; Test and evaluation equipment; Interim processor transition; F-16 MIL-STD-1589B compiler; Fairchild marketing approach; Life cycle cost analysis; MIL-STD-1750A; Managing the standard; Transition plan to upgrade the digital integrating subsystem to MIL-STD-1750A for MRASM; wasp weapon system/digital processing; Delco electronics MIL-STD-1750A architecture computer family; A new silicon-on-sapphire MIL-STD-1750A microprocessor; Texas Instruments VHSIC MIL-STD-1750A microprocessor; Testability support system for 1750A; In-circuit fault simulator; Embedded computer standardization program office (ECSPD); An implementation of a 1750 computer and its working environment; Support of software tools; Use of the 1750A support software on an IBM 3033 under MV/CMS; The AFWAL MIL-STD-1589B Jovial compiler.

-14- 412427 C.NTIS

Titre anglais : Definition, Analysis and Development of an Optical Data Distribution Network for Integrated Avionics

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

and Control Systems. Part 2: Component Development and System Integration.

Auteurs : YEN H. W.; MORRISON R. J.
 Organisme auteur: Hughes Research Labs., Malibu, CA.*National Aeronautics and Space Administration, Washington, DC.

Type Document : Report
 RPT,CTR,GRT : RPT NAS 1.26:172429; CTR NAS1-15829
 Source : Final Report, 1 May 1979 - 31 Mar. 1984.; NP. 80; NTIS Prices: PC A05/MF A01; Monitor NASA-CR-172429; DP. Jun 84

Resume : Fiber optic transmission is emerging as an attractive concept in data distribution onboard civil aircraft. Development of an Optical Data Distribution Network for Integrated Avionics and Control Systems for commercial aircraft will provide a data distribution network that gives freedom from EMI-RFI and ground loop problems, eliminates crosstalk and short circuits, provides protection and immunity from lightning induced transients and give a large bandwidth data transmission capability. In addition there is a potential for significantly reducing the weight and increasing the reliability over conventional data distribution networks. Wavelength Division Multiplexing (WDM) is a candidate method for data communication between the various avionic subsystems. With WDM all systems could conceptually communicate with each other without time sharing and requiring complicated coding schemes for each computer and subsystem to recognize a message. However, the state of the art of optical technology limits the application of fiber optics in advanced integrated avionics and control systems. Therefore, it is necessary to address the architecture for a fiber optics data distribution system for integrated avionics and control systems as well as develop prototype components and systems.

-15- 409485 C.NTIS

Titre anglais : Software Control and System Configuration Management: A Systems-Wide Approach.

Ref. Conference : Previously Announced in Iaa as A84-26713. Prepared in Cooperation with NASA. Ames Research Center, Moffett Field, Calif. Presented at IEEE/Aiaa 5TH Digital Avionics Systems Conf., Seattle, 31 Oct. - 3 Nov. 1983.

Auteurs : PETERSEN K. L.; FLORES C.
 Organisme auteur: National Aeronautics and Space Administration, Edwards, CA. Hugh L. Dryden Flight Research Center.

Type Document : Conference
 RPT,CTR,GRT : RPT NAS 1.15:85908; RPT H-1256
 Source : NP. 21; NTIS Prices: PC A02/MF A01; Monitor NASA-TN-85908; DP. Aug 84

Resume : A comprehensive software control and system configuration management process for flight-critical digital control systems of advanced aircraft has been developed and refined to insure efficient flight system development and safe flight operations. Because of the highly complex interactions among the hardware, software, and system elements of state-of-the-art digital flight control system designs, a systems-wide approach to configuration control and management has been used. Specific procedures are implemented to govern discrepancy reporting and reconciliation, software and hardware change control, systems verification and validation testing, and formal documentation requirements. An active and knowledgeable configuration control board reviews and approves all flight system configuration modifications and revalidation tests. This flexible process has proved effective during the development and flight testing of several research aircraft and remotely piloted research vehicles with digital flight control

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

systems that ranged from relatively simple to highly complex, integrated mechanizations.

-16- 409482 C.NTIS

Titre anglais : Software Modifications to the Demonstration Advanced Avionics Systems (DAAS).
Auteurs : NEDELL B.; HARDY G.
Organisme auteur: National Aeronautics and Space Administration, Moffett Field, CA. Ames Research Center.
Type Document : Report
RPT,CTR,GRT : RPT NAS 1.15:85942; RPT A-9713
Source : NP. 36; NTIS Prices: PC A03/MF A01; Monitor NASA-TM-85942; DP. Aug 84
Resume : Critical information required for the design of integrated avionics suitable for generation aviation is applied towards software modifications for the Demonstration Advanced Avionics System (DAAS). The program emphasizes the use of data busing, distributed microprocessors, shared electronic displays and data entry devices, and improved functional capability. A demonstration advanced avionics system (DAAS) is designed, built, and flight tested in a Cessna 402, twin engine, general aviation aircraft. Software modifications are made to DAAS at Ames concurrent with the flight test program. The changes are the result of the experience obtained with the system at Ames, and the comments of the pilots who evaluated the system.

-17- 401600 C.NTIS

Titre anglais : Fault Tolerant Architectures for Integrated Aircraft Electronics Systems, Task 2.
Auteurs : LEVITT K. N.; MELLIAR SMITH P. M.; SCHWARTZ R. L.
Organisme auteur: SRI International, Menlo Park, CA.*National Aeronautics and Space Administration, Washington, DC.
Type Document : Report
RPT,CTR,GRT : RPT NAS 1.26:172282; CTR NAS1-17067
Source : Final Report.; NP. 96; NTIS Prices: PC A05/MF A01; Monitor NASA-CR-172282; DP. Jun 84
Resume : The architectural basis for an advanced fault tolerant on-board computer to succeed the current generation of fault tolerant computers is examined. The network error tolerant system architecture is studied with particular attention to intercluster configurations and communication protocols, and to refined reliability estimates. The diagnosis of faults, so that appropriate choices for reconfiguration can be made is discussed. The analysis relates particularly to the recognition of transient faults in a system with tasks at many levels of priority. The demand driven data-flow architecture, which appears to have possible application in fault tolerant systems is described and work investigating the feasibility of automatic generation of aircraft flight control programs from abstract specifications is reported.

-18- 397875 C.NTIS

Titre anglais : Quantum Leap in Avionics.
Ref. Conference : This article is from 'Proceedings Papers of the AFSC (Air Force Systems Command) Avionics Standardization Conference (2nd) Held at Dayton, Ohio on 30 November-2 December 1982. Volume 2', AD-A142 777, p959-974.
Auteurs : CANTRELL W. E.
Organisme auteur: General Dynamics, Fort Worth, TX. Fort Worth Div.
Type Document : Conference
Source : NP. 16; NTIS Prices: PC A02/MF A01; DP. Nov 82
Resume : Current standardization levels in such program as the F-16 are providing benefits of productivity and growth that have been significant in the success of that program. The ever-increasing drive to performance, multi-use systems and diverse weapons has heavily taxed current avionic resources. In

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

addition, the data transfer requirement is complicated by the high speed data flow that modern computers both feed on and produce; by multiple source-multiple destination video distribution requirement; the need to self-test the system to lower levels; and the desire to dynamically reconfigure from a failure. Fortunately, the technology to achieve solutions to these new problems is evolving in the VHSIC and fiber optics programs, so that it is possible to rearchitecture the system at the module level as opposed to the LRU level. Module level standardization around a small number of types allows a large number of system level combinations while achieving economies of scale at the module level. The usual objection to standardization, that it freezes innovation, is avoided by technology transparency provisions; while at the same time the objection that standardization obsoletes the present is avoided by downward compatibility provisions. Candidates for standardization in this approach include bus interfaces, the system network, modules and racks. (Author).

-19- 397859 C.NTIS

Titre anglais : PAVE PILLAR: A Maturation Process for an Advanced Avionics Architecture.
Ref. Conference : This article is from 'Proceedings Papers of the AFSC (Air Force Systems Command) Avionics Standardization Conference (2nd) Held at Dayton, Ohio on 30 November-2 December 1982. Volume 2', AD-A142 777, p675-690.
Auteurs : MORGAN D. R.; BELLEM R. D.
Organisme auteur : Air Force Wright Aeronautical Labs., Wright-Patterson AFB, OH.
Type Document : Conference
Source : NP. 16; NTIS Prices: PC A02/MF A01; DP. Nov 82
Resume : Recent speed and density advancements in microelectronics will now permit the development of powerful and affordable avionic architectural elements - viz. processing, memories and wide band data buses. An advanced architecture making use of these elements and coupled with highly flexible software, will enhance the capability to fully exploit information integration and automation processes. An abundance of real-time data is available for integration from diverse subsystems aboard advanced military aircraft. Dramatic improvements in avionic system availability, crew workload reduction, weapon system survivability and supportability are possible using this approach. The introduction of these system integration technologies into the force structure is needed at the earliest opportunity to meet expanding mission requirements. However, care must be exercised to ensure that concepts and standards have been matured through validation testing to avoid potentially costly mistakes. The Air Force has established the PAVE PILLAR Program to provide the need maturation of advanced avionic system architectural approaches, system elements and potential system standards during advanced development. This Program also initiates the concept of establishing system integration technology as a separate discipline within the Laboratory framework. This paper describes the PAVE PILLAR Program being pursued within the Avionics Laboratory of the Wright Aeronautical Laboratories. Enhancements to operational force effectiveness resulting from system integration will be described, along with advanced system technology elements and potential standards which will be developed and demonstrated. (Author).

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-20- 397857 C.NTIS

Titre anglais : Elements for Successful Implementation of Computing Standards.
Ref. Conference : This article is from 'Proceedings Papers of the AFSC (Air Force Systems Command) Avionics Standardization Conference (2nd) Held at Dayton, Ohio on 30 November-2 December 1982. Volume 2', AD-A142 777, p643-653.
Auteurs : ENGLAND G. R.
Organisme auteur : General Dynamics, Fort Worth, TX. Fort Worth Div.
Type Document : Conference
Source : NP. 11; NTIS Prices: PC A02/MF A01; DP. Nov 82
Resume : The F-16 avionics implements what is likely the broadest application of standards of any USAF weapon system. Standards available in 1976 were applied which consisted of the MIL-STD-1553 Multiplex Data Bus, JOVIAL J3B which was the defacto software HOL and precursor to JOVIAL J73 dialect and the MIL-STD-483/490 software documentation standard. These standards were instrumental in making the F-16 a very successful program. The F-16 avionic system is now being greatly expanded to accommodate advanced sensors and weapons currently in USAF funded development. Once again the F-16 is at the forefront in implementing the latest USAF standards. A key feature of the enhanced avionics is the application of JOVIAL J73 (MIL-STD-1589B) for all subsystems, the MIL-STD-1553B Multiplex Data Bus, the MIL-STD-1750A Computer Instruction Set Architecture and the MIL-STD-1760 Stores Interface. This paper describes the implementation of standards in both the current and the enhanced F-16 avionics. (Author).

-21- 397837 C.NTIS

Titre anglais : Common 1553B I/O Channel for the F-16.
Ref. Conference : This article is from 'Proceedings Papers of the AFSC (Air Force Systems Command) Avionics Standardization Conference (2nd) Held at Dayton, Ohio on 30 November-2 December 1982. Volume 1', AD-A142 776, p367-382.
Auteurs : ALFORD S.
Organisme auteur : General Dynamics, Fort Worth, TX. Fort Worth Div.
Type Document : Conference
Source : NP. 16; NTIS Prices: PC A02/MF A01; DP. Nov 82
Resume : The 1980's will see increased standardization in military avionics. MIL-STD-1553 has proven to be an effective means of assuring communications among independently developed avionic subsystems. Future applications of the Air Force standard computer architecture. MIL-STD-1750A, and standard programming language, MIL-STD-1589B, will further decrease the life cycle costs of many systems currently under development. While MIL-STD-1750A defines a specific CPU architecture, and MIL-STD-1553B defines the method of communication among subsystems, no current Air Force standard defines the I/O channel that links the 1750 processor to the 1553 bus. In order to reduce both the cost of software development and maintenance, General Dynamics has developed a 1553 channel architecture to be applied to all the subsystems being programmed in-house for the F-16 Multi-National Staged Improvement Program (MSIP). (Author).

-22- 397815 C.NTIS

Titre anglais : Multiple System OFF Support (MSOS) System, a Pre-PMRT Capability for Evaluating Tactical Software.
Ref. Conference : This article is from 'Proceedings Papers of the AFSC (Air Force Systems Command) Avionics Standardization Conference (2nd) Held at Dayton, Ohio on 30 November-2 December 1982. Volume 1', AD-A142 776, p85-92.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Auteurs : KIRCHOFF M.; VAJO V.; LOWERY H.
 Organisme auteur: Intermetrics, Inc., Huntington Beach, CA.
 Type Document : Conference
 Source : NP. 8; NTIS Prices: PC A02/MF A01; DP. Nov 82
 Resume : A generic software testing facility is presently under development at Warner-Robins Air Logistics Command. The Multiple-System GPP Support System will allow independent verification and validation of avionics software for a variety of systems to be conducted early in the development cycle, reducing costs to the Air Force. Intermetrics, Inc. is linking via hardware and associated software the Nanodata OM-1 microprogrammable computer and the VAX-11/780. The AM-1 hosts emulations of tactical embedded computers and the VAX hosts simulations of real environments. Overlaying the emulation/simulation system in a UNIX-based monitor tailored to provide absolute control and complete visibility into the executing target machine software. A variety of static test tools for analyzing JOVIAL and, eventually, Ada code is being hosted on the VAX. A primary function of these tools will be to verify the conformance of the code to the specific standards. (Author).

-23- 378338 C.NTIS

Titre anglais : Development of a Redundant Computer System for Flight Augmentation.
 Auteurs : HERRSCHER G.; KIRST B.; SCHMIDT D.; SZLACHTA J.
 Organisme auteur: Litton Technische Werke, Freiburg im Breisgau (Germany, F.R.).*National Aeronautics and Space Administration, Washington, DC.
 Type Document : Report
 RPT,CTR,GRT : RPT BMFT-FB-W-83-027
 Source : Final Report, Apr. 1983.; In German; English Summary. Sponsored by Bundesministerium fuer Forschung und Technologie.; NP. 130; NTIS Prices: PC A07/MF A01; DP. Oct 83
 Resume : A fault-tolerant airborne computer system for time-critical process control applications is described. It guarantees real-time processing continuity in the event of hardware failures. The redundancy is hierarchically graded. The structure comprises a bus-coupled homogenous multicomputer system based on a redundant modular design. Fault detection techniques, fault diagnosis, reconfiguration, and development of process control and communication in a decentrally organized multiprocessor system are discussed. The design of a functional model and reliability estimates are outlined.

-24- 375792 C.NTIS

Titre anglais : Development and Analysis of the Software Implemented Fault-Tolerance (SIFT) Computer.
 Auteurs : GOLDBERG J.; KAUTZ W. H.; MELLIER SMITH P. M.; GREEN M. W.; LEVITT K. N.
 Organisme auteur: SRI International, Menlo Park, CA.*National Aeronautics and Space Administration, Washington, DC.
 Type Document : Report
 RPT,CTR,GRT : RPT NAS 1.26:172146; CTR NAS1-15428
 Source : Final Report.; NP. 209; NTIS Prices: PC A10/MF A01; Monitor NASA-CR-172146; DP. Feb 84
 Resume : SIFT (Software Implemented Fault Tolerance) is an experimental, fault-tolerant computer system designed to meet the extreme reliability requirements for safety-critical functions in advanced aircraft. Errors are masked by performing a majority voting operation over the results of identical computations, and faulty processors are removed from service by reassigning computations to the nonfaulty processors. This scheme has been implemented in a special architecture using a set of standard Bendix BDX930 processors, augmented by a

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

special asynchronous-broadcast communication interface that provides direct, processor to processor communication among all processors. Fault isolation is accomplished in hardware; all other fault-tolerance functions, together with scheduling and synchronization are implemented exclusively by executive system software. The system reliability is predicted by a Markov model. Mathematical consistency of the system software with respect to the reliability model has been partially verified, using recently developed tools for machine-aided proof of program correctness.

-25- 374287 C.NTIS

Titre anglais : Integration of ICNIA (Integrated Communications Navigation and Identification Avionics) into Advanced High Performance Fighter Aircraft.
Ref. Conference : This article is from 'Advanced Concepts for Avionics/Weapon System Design, Development and Integration: Conference Proceedings of the Avionics Panel Symposium (45th) Held at Ottawa, Canada on 18-22 April 1983,' AD-A138 600, p26-1-26-8.
Auteurs : CONRAD E. R.
Organisme auteur : General Dynamics, Fort Worth, TX. Fort Worth Div.
Type Document : Conference
Source : NP. 8; NTIS Prices: PC A02/MF A01; DP. Oct 83
Resume : The use of ICNIA will significantly improve the avionics suites of military aircraft. Advantages of ICNIA include: Reduction in space, weight, power and cooling requirements; Increase in reliability and maintainability; Decrease in life cycle cost; Ease of integration into an avionics suite via a multiplex bus; and Reconfigurability. To take advantage of these features, certain design guidelines should be followed. The basic guideline is that the airframer should control the integration of any subsystem into his avionics suite. This implies that the ICNIA interface software, and possibly some of the hardware, must be in accordance with the integration philosophies of the host platform. These philosophies will vary from one accordance with the integration philosophies of the host platform to the other. General Dynamics has an integration and partitioning concept which has functioned exceptionally well on the F-16. Each subsystem should perform its entire task and the interfaces between subsystems should be as simple as possible. This concept has three major advantages: Changes to one subsystem are transparent to other subsystems and do not result in changes being required in other subsystems when one subsystem is changed; The integration of a new subsystem into the avionics suite is not difficult since a new system does not require unique support of other subsystems; and Fault isolation is simple since each subsystem performs an entire task.

-26- 374286 C.NTIS

Titre anglais : Practical Approach to the Design of a New Avionic System.
Ref. Conference : This article is from 'Advanced Concepts for Avionics/Weapon System Design, Development and Integration: Conference Proceedings of the Avionics Panel Symposium (45th) Held at Ottawa, Canada on 18-22 April 1983,' AD-A138 600, p25-1-25-11.
Auteurs : DUKE P. A.
Organisme auteur : British Aerospace Public Ltd., Co., Brough (England). Kingston-Brough Div.
Type Document : Conference
Source : NP. 11; NTIS Prices: PC A02/MF A01; DP. Oct 83
Resume : This paper describes a program to design, construct and demonstrate an advanced avionic system for the next generation of tactical combat aircraft. British Aerospace is carrying out such a program with the objective of reducing development risks associated

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

with the rapid advance of technology. A number of factors contribute to this risk, notably the dramatic increase in system capability made possible by the general availability of LSI and VLSI circuitry. This has occurred at a time when the next aircraft project is likely to have a single seat cockpit. Traditionally independent systems can now be linked using a data bus to provide a fully integrated system with the pilot's needs foremost in mind. The system is based on a multi bus architecture recognising the differing integrity requirements of different parts of the system. A complete aircraft system is represented, divided into functional groups, and includes the basic aircraft systems such as hydraulics and fuel management and an integral maintenance reporting system. Mission systems include a wide range of sensor and weapon types. The practical implications of introducing Mil.Std.1760 or the associated STANAG 3837AA standard store interfaces are being studied. The avionic systems are linked to an advanced cockpit, with the objective of reducing pilot workload. The cockpit makes use of multi-purpose displays and an integrated approach to system control. A display of the out-of-cockpit scene is provided to allow the 'pilot' to operate the controls in a realistic manner and so provide representative input to the avionic system.

-27- 374281 C.NTIS

Titre anglais : Network Communications for a Distributed Avionics System.
Ref. Conference : This article is from 'Advanced Concepts for Avionics/Weapon System Design, Development and Integration: Conference Proceedings of the Avionics Panel Symposium (45th) Held at Ottawa, Canada on 18-22 April 1983,' AD-A138 600, p19-1-19-10.
Auteurs : DSTGAARD J. C.; ZANN D. A.
Organisme auteur : Air Force Avionics Lab., Wright-Patterson AFB, OH.
Type Document : Conference
Source : NP. 10; NTIS Prices: PC A02/MF A01; DP. Oct 83
Resume : Due to the postulated 1990's threat environment advanced avionics architectures are experiencing demands for increased performance which have led, in part, to increased processing requirements and system complexity. As more processors are added to the control environment of sophisticated military aircraft, the choice of processor interconnection topology, and methodology assumes greater importance. This choice profoundly influences information throughput, reliability, survivability and integrity throughout the weapon system. The ability to rapidly exchange/transfer information among processors and devices is critical if one is to develop a reliable, effective communication system. This paper addresses basic communication techniques which could serve as candidates in satisfying the network communication requirements of an advanced avionics architecture. Features of each technique are examined to ascertain the performance of these multi-access protocols in terms of developed system-driven criteria. (Author).

-28- 374267 C.NTIS

Titre anglais : System Architecture: Key to Future Avionics Capabilities.
Ref. Conference : This article is from 'Advanced Concepts for Avionics/Weapon System Design, Development and Integration: Conference Proceedings of the Avionics Panel Symposium (45th) Held at Ottawa, Canada on 18-22 April 1983,' AD-A138 600, p1-1-1-6.
Auteurs : ENGLAND G. R.
Organisme auteur : General Dynamics, Fort Worth, TX. Fort Worth Div.
Type Document : Conference
Source : NP. 6; NTIS Prices: PC A02/MF A01; DP. Oct 83

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Resume : Appropriate architectural approaches in the physical, functional, information transfer, and control system areas are the key to future avionic capabilities. The appropriate architectures will provide dramatic improvements in system performance while simultaneously improving system availability, supportability, and affordability. The software and hardware technology required to achieve these architectural improvements are already here and simply need to be improved, properly applied, and integrated. The resulting weapon systems will, however, have widespread effects in many areas of operations, logistics, and equipment acquisition. Changes will be required in the way pilots are trained and conduct their missions. The proper pilot/vehicle interface will need to be developed to fully allow the pilot to act as a system manager. At the same time data must be provided to assure pilot confidence that the automated system is accomplishing properly the detailed operational tasks which were formerly accomplished manually. Procurement of avionic systems and spares will undergo a dramatic change. Common modules will likely to be procured directly by the military from software and hardware module sources and will be provided to avionic vendors. Avionic systems developers will find themselves creating special sensor and effector modules and function-unique software to be used with modules common to many other uses. With large numbers of throwaway modules, depot repair facilities and organizations will shrink, or the function will revert to the original manufacturer.

-29- 371079 C.NTIS

Titre anglais : Study of Optimal Computer Network Architecture for Digital Avionics Systems.
Auteurs : KRILIC M. F.
Organisme auteur : Air Force Inst. of Tech., Wright-Patterson AFB, OH. School of Engineering.
Type Document : Thesis
RPT,CTR,GRT : RPT AFIT/GE/EE/83D-36
Source : Master's thesis; NP. 221; NTIS Prices: PC A10/MF A01; DP. Dec 83

Resume : In this paper, current theoretical considerations in available literature have been used to sort out the essential figures of merit of computer network architectures for digital avionic systems. Fourteen different approaches to the same problem of data multiplexing in avionics systems are analyzed according to the key issues. Conclusions drawn are used to define the optimal computer network architecture for digital avionics. The Self-Managing Multiplex System (SMS) is conceptually designed with respect to the optimal characteristics, along with the discussions of some trade-offs that had to be made. The burst errors self-correcting feature of the broadcast-acknowledgements in the SMS concept seems to deserve some sort of testing in practice. It is recommended that a detailed simulation study should be performed later and a hot bench built up using the latest technologies that exist. (Author).

-30- 362558 C.NTIS

Titre anglais : Design Implications from Afti/F-16 Flight Test.
Ref. Conference : Presented at Iee/Aiaa 5TH Digital Avionics Systems Conf., Seattle, 31 Oct. - 3 Nov. 1983.
Auteurs : ISHMAEL S. D.; REGENIE V. A.; MACKALL D. A.
Organisme auteur : National Aeronautics and Space Administration, Moffett Field, CA. Ames Research Center.
Type Document : Conference
RPT,CTR,GRT : RPT NAS 1.15:86026; RPT H-1213
Source : Final Report.; NP. 13; NTIS Prices: PC A02/MF A01; Monitor NASA-TM-86026; DP. Jan 84
Resume : Advanced fighter technologies are evolving into

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

highly complex systems. Flight controls are being integrated with advanced avionics to achieve a total system. The advanced fighter technology integration (AFTI) F-16 aircraft is an example of a highly complex digital flight control system integrated with advanced avionics and cockpit. The architecture of these new systems involves several general issues. The use of dissimilar backup modes if the primary system fails requires the designer to trade off system simplicity and capability. This tradeoff is evident in the AFTI/F-16 aircraft with its limited stability and fly-by-wire digital flight control systems. In case of a generic software failure, the backup or normal mode must provide equivalent envelope protection during the transition to degraded flight control. The complexity of systems like the AFTI/F-16 system defines a second design issue, which can be divided into two segments: the effect on testing, and the pilot's ability to act correctly in the limited time available for cockpit decisions. The large matrix of states possible with the AFTI/F-16 flight control system illustrates the difficulty of both testing the system and choosing real-time pilot actions.

-31- 350142 C.NTIS

Titre anglais : Fault Tolerant Architectures for Integrated Aircraft Electronics Systems.
Auteurs : LEVITT K. N.; MELLIER SMITH P. M.; SCHWARTZ R. L.
Organisme auteur: SRI International, Menlo Park, CA.*National Aeronautics and Space Administration, Washington, DC.
Type Document : Report
RPT,CTR,GRT : RPT NAS 1.26:172226; CTR NAS1-17067; CTR SRI PROJ. 4616
Source : Final Report.; NP. 57; NTIS Prices: PC A04/MF A01; Monitor NASA-CR-172226; DP. Aug 83
Resume : Work into possible architectures for future flight control computer systems is described. Ada for Fault-Tolerant Systems, the NETS Network Error-Tolerant System architecture, and voting in asynchronous systems are covered.

-32- 284125 C.NTIS

Titre anglais : Advanced Flight Control System Study.
Auteurs : HARTMANN G. L.; WALL J. E.; RANG E. R.; LEE H. P.; SCHULTE R. W.
Organisme auteur: Honeywell Systems and Research Center, Minneapolis, MN.*National Aeronautics and Space Administration, Washington, DC.
Type Document : Report
RPT,CTR,GRT : RPT NAS 1.26:163117; RPT HONEYWELL-82SRC5; CTR NAS4-2876
Source : Final Report.; Prepared in Cooperation with Lockheed-California CO., Burbank.; NP. 171; NTIS Prices: PC A08/MF A01; Monitor NASA-CR-163117; DP. Nov 82
Resume : A fly by wire flight control system architecture designed for high reliability includes spare sensor and computer elements to permit safe dispatch with failed elements, thereby reducing unscheduled maintenance. A methodology capable of demonstrating that the architecture does achieve the predicted performance characteristics consists of a hierarchy of activities ranging from analytical calculations of system reliability and formal methods of software verification to iron bird testing followed by flight evaluation. Interfacing this architecture to the Lockheed S-3A aircraft for flight test is discussed. This testbed vehicle can be expanded to support flight experiments in advanced aerodynamics, electromechanical actuators, secondary power systems, flight management, new displays, and air traffic control concepts.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-33- 273565 C.NTIS

Titre anglais : Demonstration Advanced Avionics System (Daas)
Function Description.
Auteurs : BAILEY A. J.; BAILEY D. G.; GAABO R. J.; LAHN T. G.;
LARSON J. C.
Organisme auteur: Honeywell, Inc., Minneapolis, MN.*National
Aeronautics and Space Administration, Washington,
DC.
Type Document : Report
RPT,CTR,GRT : RPT NAS 1.26:166282; CTR NAS2-10021
Source : Prepared in Cooperation with King Radio Corp.,
Olathe, Kans.; NP. 204; NTIS Prices: PC A10/MF A01;
Monitor NASA-CR-166282; DP. Jan 82
Resume : The Demonstration Advanced Avionics System, DAAS, is
an integrated avionics system utilizing
microprocessor technologies, data busing, and shared
displays for demonstrating the potential of these
technologies in improving the safety and utility of
general aviation operations in the late 1980's and
beyond. Major hardware elements of the DAAS include
a functionally distributed microcomputer complex, an
integrated data control center, an electronic
horizontal situation indicator, and a radio adaptor
unit. All processing and display resources are
interconnected by an IEEE-488 bus in order to
enhance the overall system effectiveness,
reliability, modularity and maintainability. A
detail description of the DAAS architecture, the
DAAS hardware, and the DAAS functions is presented.
The system is designed for installation and flight
test in a NASA Cessna 402-B aircraft.

-34- 250705 C.NTIS

Titre anglais : An Assessment of the Real-Time Application
Capabilities of the SIFT Computer System.
Auteurs : BUTLER R. W.
Organisme auteur: National Aeronautics and Space Administration,
Hampton, VA. Langley Research Center.
Type Document : Report
RPT,CTR,GRT : RPT NASA-TM-84482; RPT NAS 1.15:84482
Source : NP. 17; NTIS Prices: PC A02/MF A01; DP. Apr 82
Resume : The real-time capabilities of the SIFT computer
system, a highly reliable multicomputer architecture
developed to support the flight controls of a
relaxed static stability aircraft, are discussed.
The SIFT computer system was designed to meet
extremely high reliability requirements and to
facilitate a formal proof of its correctness.
Although SIFT represents a significant achievement
in fault-tolerant system research it presents an
unusual and restrictive interface to its users. The
characteristics of the user interface and its impact
on application system design are assessed.

-35- 242687 C.NTIS

Titre anglais : Guidance and Control Technology for Highly
Integrated Systems.
Organisme auteur: Advisory Group for Aerospace Research and
Development, Neuilly-sur-Seine (France). *National
Aeronautics and Space Administration, Washington,
DC.
Type Document : Report
RPT,CTR,GRT : RPT AGARD-CP-314
Source : In English and French. the 33RD Symp. Held in
Athens, 13-16 Oct. 1981.; NP. 179; NTIS Prices: PC
A09/MF A01; DP. Feb 82
Resume : No abstract available.

-36- 223298 C.NTIS

Titre anglais : Survivable Avionics Computer System.
Auteurs : MONSON P. R.; MONSON C. A.; PEASE M. C.; WISCHMEYER
C. E.
Organisme auteur: SRI International, Menlo Park, CA.
Type Document : Report
RPT,CTR,GRT : CTR F33615-80-C-1014

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Source : Final rept.; NP. 241; NTIS Prices: PC A11/MF A01; DP. Nov 80

Resume : This contract has shown by the examples of navigation and flight control that the CHAMP approach to architecture design can be applied to avionic computational problems. Furthermore, the simulations have shown that CHAMP can provide a potentially survivable computer for aircraft use. CHAMP uses a simple network which can make use of advanced hardware techniques (such as those available through the VHSIC program). It also simplifies material inventory problems since there is only one type of spare (a single PC) which is contained in a single chip. The PC used in this study is well within the capability of single-chip techniques projected for DAIS the 1985 time period. The avionics functions software used in this study were created by using DAIS project personnel structured programming techniques which yielded sub-modules for each function that were small and tractable and easily managed by microprocessor-based computers of modest capability. This is especially significant considering that this can be accomplished with processors operating at relatively low-frequency clock rates (1 MHz as opposed to the 10 to 20-MHz clock often used in more elaborate processors). This results in a much more reliable and easily constructed hardware module for the computer. Furthermore, this relative simplicity tends to make the computing modules immune to noise, interference, and EMP.

-37- 213697 C.NTIS

Titre anglais : Tactical Airborne Distributed Computing and Networks.

Ref. Conference : Presented at a Meeting of the Avionics Panel Held in Roeros, Norway, 22-25 Jun 81.

Organisme auteur : Advisory Group for Aerospace Research and Development, Neuilly-sur-Seine (France).

Type Document : Conference

RPT,CTR,GRT : RPT AGARD-CP-303

Source : Conference proceedings.; NP. 414; NTIS Prices: PC A18/MF A01; DP. Oct 81

Resume : These proceedings consist of the papers and discussions. The 35 papers were divided as follows, three on state-of-the-art; five on system architecture; four on system design approaches; five on software; five on fault tolerance and reliability; six on interconnection, bussing and networking; seven on applications to avionics systems.

-38- 208935 C.NTIS

Titre anglais : Digital Avionics Information System (DAIS): Development and Demonstration.

Auteurs : COOK M. J.; MASON R. C.; STAUTBERG J. L.; SELF L. E.; ELLISON R. L.

Organisme auteur : TRW Defense and Space Systems Group, Redondo Beach, CA.*Air Force Wright Aeronautical Labs., Wright-Patterson AFB, OH.

Type Document : Report

RPT,CTR,GRT : CTR F33615-78-C-1502

Source : Final rept. 1 Oct 78-31 Jul 81; NP. 201; Proj. 2052; Task 05; NTIS Prices: PC A10/MF A01; Monitor AFWAL-TR-81-1165; DP. Sep 81

Resume : The Digital Avionics Information System (DAIS) represents a significant advance in the technology of avionics system architecture. DAIS is a total systems concept, exploiting standardization, modularity, and application independent executive software to provide a system architecture adaptable to many aircraft, missions, and avionics configurations and fully capable of accommodating new advances in technology. These fundamental system characteristics are described in this report; the

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

specific system features which provide these
characteristics and attributes are presented

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

- 1- 836783 C.INSPEC
 Titre Anglais : THE IMPACT OF ADVANCED COMPUTER SYSTEMS ON AVIONICS
 RELIABILITY.
 Auteurs : BORDEN A. G.
 Type Document : Journal Article
 Titre Journal : DEF. ELECTRON. (USA)
 Source : VOL.19, NO.5; PP. 57-8, 10, 14, 17-18, 21; REF. 12;
 DP MAY 1987
 Coden : DEELDH
 Issn : 0194-7885
 Resume : Integrated avionics systems like pave pillar and
 icnia are among the many beneficiaries of advanced
 computer architectures and vhsic technology.
- 2- 819377 C.INSPEC
 Titre Anglais : MODULAR ICNIA PACKAGING TECHNOLOGY.
 Auteurs : PORADISH F.
 Affiliation : TEXAS INSTRUMENTS, MCKINNEY, TX, USA
 Type Document : Journal Article
 Titre Journal : IEEE AEROSP. AND ELECTRON. SYST. MAG. (USA)
 Source : VOL.2, NO.6; PP. 20-3; REF. 2; CCCC=
 0885-8985/87/0600-0020 DOLLARS 01.00; DP JUNE 1987
 Coden : IESMEA
 Issn : 0885-8985
 Resume : The author discusses the integrated communication
 navigation identification avionics program, which is
 achieving significant size, weight, power, and
 reliability improvements by the modular integration
 of similar functions into a fault-tolerant
 reconfigurable architecture. This is being
 accomplished with a combination of modular circuit
 designs incorporating surface-mount component
 technology, and a modular two-level maintenance
 support concept for reduced life cycle cost. The
 focus is on the modular packaging technology of the
 digital processor subsystem.
- 3- 780666 C.INSPEC
 Titre Anglais : THE APPLICABILITY OF THE EMERGING AMERICAN NATIONAL
 STANDARD FIBER DISTRIBUTED DATA INTERFACE (FDDI) FOR
 A DISTRIBUTED AVIONICS ARCHITECTURE.
 Titre Conference : DIGEST OF PAPERS, COMPCON SPRING '87, THIRTY-SECOND
 IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE.
 INTELLECTUAL LEVERAGE (CAT. NO.87CH2409-1)
 Lieu Conference : SAN FRANCISCO, CA, USA
 Date conference : 23-27 FEB. 1987
 Auteurs : COHN M.
 Type Document : Conference Paper
 Editeur : IEEE COMPUT. SOC. PRESS, WASHINGTON, DC, USA
 Source : NP. XVI-470; PP. 448-51; REF. 9; SPD. IEEE; CCCC=
 CH2409-1/87/0000-0448 DOLLARS 01.00; DP 1987
 Isbn : 0-8186-0764-5
 Resume : The data communication network requirements for the
 next-generation distributed avionics architecture
 are analyzed. The emerging american national
 standard fiber distributed data interface (fddi)
 local area network (lan) standard is assessed for
 its suitability in meeting those requirements. The
 consideration include: communications architecture,
 performance, and fault tolerance.
- 4- 779435 C.INSPEC
 Titre Anglais : SIMULATION MODEL OF A HIGH-SPEED TOKEN-PASSING BUS
 FOR AVIONICS APPLICATIONS.
 Titre Conference : PROCEEDINGS OF THE IEEE/AIAA 7TH DIGITAL AVIONICS
 SYSTEMS CONFERENCE (CAT. NO.86CH2359-8)
 Lieu Conference : FORT WORTH, TX, USA
 Date conference : 13-16 OCT. 1986
 Auteurs : SPIETH J. E.; SEWARD W. D.
 Affiliation : AERONAUT. SYST. DIV., WRIGHT-PATTERSON AIR FORCE
 BASE, OH, USA
 Type Document : Conference Paper
 Editeur : IEEE, NEW YORK, USA
 Source : NP. 803; PP. 242-9; REF. 17; SPD. IEEE/AIAA; DP 1986
 Resume : The design of a protocol with performance that can

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

meet the requirements for a next-generation aviation electronics (avionics) data bus is considered. A study that developed and validated a model for simulating bus token-passing protocols for avionics applications is reported. Two algorithms were designed that reflected the timing and operation of a distributed and a centralized control token-passing protocol. The algorithms were incorporated in an overall simulation model program that included control, data collection and analysis functions. The simulation model program allows various avionics bus configurations to be defined and tested. Initial performance tests of a centralized control token-passing protocol using a configuration representative of a fighter-type aircraft bus network are described, and the performance of the two types of protocols is compared.

-5- 779420 C.INSPEC

Titre Anglais : HELICOPTER AVIONICS ARCHITECTURE FOR INTEGRATING
FLIGHT CRITICAL FUNCTIONS.
Titre Conference: PROCEEDINGS OF THE IEEE/AIAA 7TH DIGITAL AVIONICS
SYSTEMS CONFERENCE (CAT. NO.86CH2359-8)
Lieu Conference : FORT WORTH, TX, USA
Date conference : 13-16 OCT. 1986
Auteurs : OSDER S. S.
Affiliation : MCDONNELL DOUGLAS HELICOPTER CO., TEMPE, AZ, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 803; PP. 134-41; REF. 15; SPO. IEEE/AIAA; CCCC=
CH2359-8/86/0000-0134 DOLLARS 01.00; DP 1986
Resume : An approach to the mechanization of the traditional
navigation function is described that can provide
the key integration interface between the
flight-critical aircraft fly-by-wire stabilization
and control and the remainder of the mission
avionics. Redundant, integrated navigation and
sensor assemblies provide all of the aircraft
position, velocity, acceleration, angular rate,
attitude, heading, and air data states needed for
both the flight control as well as the mission
management functions. The architecture concept uses
functional partitioning with distributed processing
aimed at decoupling software dependencies between
the various 'integrated' avionics system elements.

-6- 779414 C.INSPEC

Titre Anglais : UNIVERSAL RECEIVER FOR ICNIA.
Titre Conference: PROCEEDINGS OF THE IEEE/AIAA 7TH DIGITAL AVIONICS
SYSTEMS CONFERENCE (CAT. NO.86CH2359-8)
Lieu Conference : FORT WORTH, TX, USA
Date conference : 13-16 OCT. 1986
Auteurs : SMEAD F. W.
Affiliation : ITT AVIONICS DIV., NUTLEY, NJ, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 803; PP. 85-9; SPO. IEEE/AIAA; CCCC=
CH2359-8/86/0000-0085 DOLLARS 01.00; DP 1986
Resume : The architecture for the icnia (integrated
communication navigation identification avionics)
system, under development for the us military, is
highly modular. Radio receive and transmit functions
are accomplished through real-time
computer-controlled interconnection of appropriate
modules, whose parameters are also changed in real
time, as required, for each specific signal and
signal type. With such an architecture, the more
flexible each type of module, the easier and more
efficient it is to implement the wide variety of
signal types which the icnia system must process and
to new signal types in the future. The design of
such a receiver is described.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-7- 766643 C.INSPEC

Titre Anglais : FAULT-FREE PERFORMANCE VALIDATION OF AVIONIC
MULTIPROCESSORS.
Titre Conference: PROCEEDINGS OF THE IEEE/AIAA 7TH DIGITAL AVIONICS
SYSTEMS CONFERENCE (CAT. NO.86CH2359-8)
Lieu Conference : FORT WORTH, TX, USA
Date Conference : 13-16 OCT. 1986
Auteurs : CZECK E. W.; FEATHER F. E.; GRIZZAFFI A. M.; FINELLI
G. B.; SEGALL Z. Z.;
Affiliation : CARNEGIE MELLON UNIV., PITTSBURGH, PA, USA (03);
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 803; PP. 670-7; REF. 10; SPD. IEEE/AIAA; CCCC=
CH2359-8/86/0000-0670 DOLLARS 01.00; DP 1986
Resume : The application of a validation methodology to
nasa's fault-tolerant multiprocessor system and the
software implemented fault-tolerance computer system
is described. The methodology entails a building
block approach, starting with simple baseline
experiments and building to more complex
experiments. The goal methodology is to test and
characterize thoroughly the performance and behavior
of ultrareliable computer systems. The results show
that the methodology is not machine-specific and can
be used in lieu of life-testing approaches. By
applying a building-block approach at the systems
level, the machine complexity was broken down to
manageable levels independent of system
implementation.

-8- 766641 C.INSPEC

Titre Anglais : CHANNELIZED OR NONCHANNELIZED FAULT-TOLERANT
COMPUTERS: A HARDWARE COMPLEXITY COMPARISON OF
FAULT-TOLERANT COMPUTERS FOR FLIGHT CONTROL SYSTEMS.
Titre Conference: PROCEEDINGS OF THE IEEE/AIAA 7TH DIGITAL AVIONICS
SYSTEMS CONFERENCE (CAT. NO.86CH2359-8)
Lieu Conference : FORT WORTH, TX, USA
Date conference : 13-16 OCT. 1986
Auteurs : SCHMID H.; LARIMER S.; MADAK T.
Affiliation : GENERAL ELECTRIC CO., BINGHAMTON, NY, USA (03);
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 803; PP. 655-63; REF. 8; SPD. IEEE/AIAA; CCCC=
CH2359-8/86/0000-0655 DOLLARS 01.00; DP 1986
Resume : Flight-critical control systems for future aircraft,
such as the advanced tactical fighter (atf), demand
higher reliability, greater fault tolerance, and
more computing power as well as easier maintenance
and reduced life cycle costs. The conventional
approach to implementing these systems has been to
use channelized architectures. Nonchannelized
reconfigurable multiprocessor systems (rmpps), which
have been in development for the last decade, are
said to provide greater hardware efficiency and
lower recurring costs at the price of higher
software and system complexity, and more difficult
validation and verification processes. The hardware
complexity of the rmpps is evaluated by comparing
gate and pin counts of four fault-tolerant computer
architectures. The reliability results are
presented.

-9- 766634 C.INSPEC

Titre Anglais : A FAULT TOLERANT PROCESSOR TO MEET RIGOROUS FAILURE
REQUIREMENTS AIRBORNE APPLICATIONS!
Titre Conference: PROCEEDINGS OF THE IEEE/AIAA 7TH DIGITAL AVIONICS
SYSTEMS CONFERENCE (CAT. NO.86CH2359-8)
Lieu Conference : FORT WORTH, TX, USA
Date conference : 13-16 OCT. 1986
Auteurs : LALA J. H.; ALGER L. S.; GAUTHIER R. J.; DZWONCZYK
M. J.
Affiliation : CHARLES STARK DRAPER LAB. INC., CAMBRIDGE, MA, USA
(04);
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Source : NP. 803; PP. 555-62; REF. 8; SPO. IEEE; AIAA; CCCC-CH2359-8/86/0000-0555 DOLLARS 01.00; DP 1986

Resume : There is a need for extremely-high-reliability airborne computers for applications in advanced military and civilian aircraft to carry out such traditional functions as guidance, navigation, and flight control as well as tasks associated with more advanced functions such as an electronic pilot's associate and an electronic flight engineer. A fault-tolerant computer architecture has been designed in conformance with rigorous theory for such high-reliability applications to tolerate any arbitrary failure mode of hardware components. The computer architecture, its hardware implementation, the operating system and the redundancy management software and its interfaces to external devices are described.

-10- 695535 C.INSPEC

Titre Anglais : SOME VERIFICATION TOOLS AND METHODS FOR AIRBORNE SAFETY-CRITICAL SOFTWARE.

Auteurs : HELPS K. A.

Affiliation : SMITHS IND. AEROSPACE AND DEFENCE SYST. LTD., CHELTENHAM, ENGLAND

Type Document : Journal Article

Titre Journal : SOFTWARE ENG. J. (GB)

Source : VOL.1, NO.6; PP. 248-53; REF. 6; DP NOV. 1986

Coden : SEJOED

Issn : 0268-6961

Resume : Airborne software, like many other kinds of embedded software, grows in complexity with each generation of equipment. Where the software supports safety-critical functions this can present severe verification problems. The scale of such software is often outside the scope of mathematically formal verification, and dissimilar software redundancy techniques may be inapplicable for performance reasons. A practical approach is to meet safety-critical criteria by procedurally formal verification in line with the radio technical commission for aeronautics and the european organisation for civil aviation electronics common revised (1985) guidelines on the software aspects of certification of airborne systems, using a comprehensive automated test coverage analysis and partition breach analysis system.

-11- 695534 C.INSPEC

Titre Anglais : THE EXPERIMENTAL AIRCRAFT PROGRAMME SOFTWARE TOOLSET.

Auteurs : CRONSHAW P.

Affiliation : BAE PLC., PRESTON, ENGLAND

Type Document : Journal Article

Titre Journal : SOFTWARE ENG. J. (GB)

Source : VOL.1, NO.6; PP. 236-47; REF. 11; DP NOV. 1986

Coden : SEJOED

Issn : 0268-6961

Resume : The experimental aircraft programme (eap) is a technology demonstrator project in advance of the next generation of fighter aircraft. An integrated software toolset has been established to support the development and production of software for the advanced avionic systems embedded in this aircraft. This paper presents the work undertaken and the experience gained in developing, using and supporting the eap software toolset from project initiation to final delivery of the software for flight testing. Many of the issues pertaining to the future requirements of an integrated project support environment and the rigours imposed in delivering reliable and high-quality real-time software for embedded processors are discussed.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-12- 677729 C.INSPEC

Titre Anglais : MULTIOBJECTIVE INSENSITIVE COMPUTER-AIDED DESIGN OF
AEROSPACE CONTROL SYSTEMS.
Titre Conference: CONTROL APPLICATIONS OF NONLINEAR PROGRAMMING AND
OPTIMIZATION. PROCEEDINGS OF THE FIFTH IFAC WORKSHOP
Lieu Conference : CAPRI, ITALY
Date conference : 11-14 JUNE 1985
Auteurs : SCHY A. A.; GIESY D. P.; DI PILLO G.
Affiliation : NASA LANGLEY RES. CENTER, HAMPTON, VA, USA
Type Document : Conference Paper
Editeur : PERGAMON, OXFORD, ENGLAND
Source : NP. X-209; PP. 177-88; REF. 15; SPO. IFAC; UNIV. ROME
'LA SAPIENZA'; UNIV. CALABRIA; ET AL; DP 1986

Resume : A multiobjective cad method for aircraft control
systems has been developed which can meet
requirements on disparate objectives over a set of
flight conditions, using constrained minimization
algorithms with objective functions in the
constraint vector. This paper summarizes results of
research on four increasingly sophisticated versions
of the method: the basic method; an extension which
finds pareto optimal designs which are well-balanced
in all objectives; an extension which finds
stochastic-insensitive (si) designs to minimize the
sensitivity to uncertain parameters; and a tradeoff
method which designs for a compromise between
decreased sensitivity and improved nominal objective
values.

-13- 629392 C.INSPEC

Titre Anglais : COMMON SIGNAL PROCESSOR DESIGNED FOR MULTIPLE
APPLICATIONS.
Auteurs : LEE W. H.; DENNIS C. A.; GILBERT W. L.
Affiliation : DIV. OF FEDERAL SYST., IBM, MANASSAS, VA, USA (O3);
Type Document : Journal Article
Titre journal : DEF. ELECTRON. (USA)
Source : VOL. 18, NO. 7; PP. 176-179; DP JULY 1986
Codex : DEELDH
Issn : 0194-7885
Resume : Future military electronic systems will incorporate
more and more very large-scale integration (vlsi),
and ibm's federal systems division (fsd) is a major
participant in developing the necessary
technological groundwork. One of the first fruits of
this development effort, a military signal processor
relying almost exclusively on vlsi component
technology, is being developed by fsd for the air
force's pave pillar integrated avionics program.
Called the common signal processor (csp), it will
feature an architecture flexible enough to follow
for a variety of applications and for future growth.
The csp features advanced complementary metal-oxide
semiconductor (cmos) circuitry with resolution of
individual elements down to the 1-micron level.

-14- 620647 C.INSPEC

Titre Anglais : TEXAS INSTRUMENTS VHSIC 1750A COMPUTER.
Titre Conference: PROCEEDINGS OF THE IEEE 1986 NATIONAL AEROSPACE AND
ELECTRONICS CONFERENCE, NAECON 1986 (CAT.
NO. 86CH2307-7)
Lieu Conference : DAYTON, OH, USA
Date conference : 19-23 MAY 1986
Auteurs : NORMAN L. E.
Affiliation : TEXAS INSTRUM. INC., DALLAS, TX, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 4 VOL. 1399; PP. 125-30 VOL. 1; REF. 4; SPO.
IEEE; CCCC= 0547-3578/86/0000-0125\$01.00; DP 1986
Resume : Future tactical aircraft and vehicles will carry
several different sensors including advanced flirs,
radars, ew, and (in some cases) acoustic systems.
The sensor outputs will need to be integrated and
displayed to the operator in a condensed and
prioritized manner. To fulfill this requirement, a
new generation of programmable mil-std-1750a

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

architecture hardware is needed to produce state-of-the-art processors with high throughput and 99percent fault coverage at the module level. A description is given of new vhsic ics being used in the design of a 1750a architecture computer during 1986. The vhsic 1750a computer will provide high processing throughput, system reconfigurability, and two-level maintenance, and will lower system power requirements at significant hardware savings.

-15- 620646 C.INSPEC

Titre Anglais : CAPABILITY ASSESSMENT IN AIRBORNE PLATFORMS.
Titre Conference: PROCEEDINGS OF THE IEEE 1986 NATIONAL AEROSPACE AND ELECTRONICS CONFERENCE, NAECON 1986 (CAT. NO.86CH2307-7)

Lieu Conference : DAYTON, OH, USA

Date conference : 19-23 MAY 1986

Auteurs : ELENICAL G.; LUNDE D.

Affiliation : WESTINGHOUSE ELECTR. CORP., BALTIMORE, MD, USA (02);

Type Document : Conference Paper

Editeur : IEEE, NEW YORK, USA

Source : NP. 4 VOL. 1399; PP. 119-24 VOL.1; SPO. IEEE; CCCC=0547-3578/86/0000-0119\$01.00; DP 1986

Resume : The multimode weapons systems of the future will be highly fault-tolerant, possessing the ability to perform tactical missions with both full or degraded functional capabilities. The fault-tolerant system characteristics will allow systems with less than the fully specified functional capabilities to be engaging in combat. This design feature will present the operators of these weapons systems with the operational challenge of selecting and/or assigning weapons platforms with degraded capabilities to carry out tactical missions. An 'in-system' assessment process is proposed to evaluate the operability for these weapons platforms based on the current functional status, the reliability of the hardware resources within the system's avionics, and the resources required by the various application modes to accomplish mission tasks.

-16- 612197 C.INSPEC

Titre Anglais : AF MULTIPROCESSOR FLIGHT CONTROL ARCHITECTURE

DEVELOPMENTS: CRMMFCS AND BEYOND.
Titre Conference: PROCEEDINGS OF THE IEEE 1986 NATIONAL AEROSPACE AND ELECTRONICS CONFERENCE, NAECON 1986 (CAT. NO.86CH2307-7)

Lieu Conference : DAYTON, OH, USA

Date conference : 19-23 MAY 1986

Auteurs : THOMPSON D. B.; BORTNER R. A.

Affiliation : US AIR FORCE WRIGHT AERONAUT. LABS.,

WRIGHT-PATTERSON AIR FORCE BASE, OH, USA (02);

Type Document : Conference Paper

Editeur : IEEE, NEW YORK, USA

Source : NP. 4 VOL. 1399; PP. 376-82 VOL.2; REF. 14; SPO. IEEE; DP 1986

Resume : Since 1978, the air force wright aeronautical laboratories flight dynamics laboratory has been involved in in-house research into fault-tolerant multiprocessor flight control systems. The emphasis of these studies has been on applying microprocessor technology, dynamic sparing, and distributed parallel processing techniques rather than conventional triple or quadruple channel computer structures. The pertinent issues, systems, and concepts leading to the current work on the advanced multiprocessor control architecture definition (amcad) program are discussed.

-17- 612196 C.INSPEC

Titre Anglais : ADVANCED INFORMATION PROCESSING SYSTEM: STATUS REPORT.

Titre Conference: PROCEEDINGS OF THE IEEE 1986 NATIONAL AEROSPACE AND ELECTRONICS CONFERENCE, NAECON 1986 (CAT. NO.86CH2307-7)

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Lieu Conference : DAYTON, OH, USA
 Date conference : 19-23 MAY 1986
 Auteurs : BROCK L. D.; LALA J.
 Affiliation : CHARLES STARK DRAPER LAB. INC., CAMBRIDGE, MA, USA
 (02);
 Type Document : Conference Paper
 Editeur : IEEE, NEW YORK, USA
 Source : NP. 4 VOL. 1399; PP. 368-75 VOL.2; REF. 2; SPD.
 IEEE; CCCC= 0547-3578/86/0000-0368\$01.00; DP 1986
 Resume : The advanced information processing system (aips) is
 designed to provide a fault-tolerant and
 damage-tolerant data processing architecture for a
 broad range of aerospace vehicles. The aips
 architecture also has attributes that enhance system
 effectiveness, such as graceful degradation, growth
 and change tolerance, and integrability. Two key
 building blocks being developed for the aips program
 are a fault-and-damage tolerant processor and
 communication network. A proof-of-concept system is
 now being built and will be tested to demonstrate
 the validity and performance of the aips concepts.

-18- 597848 C.INSPEC

Titre Anglais : POSSIBLE IMPACT OF VHSIC ON MIL-STD-1553B DATA
 TRANSMISSION MANAGEMENT.
 Titre Conference: IMPACT OF VERY HIGH PERFORMANCE INTEGRATED CIRCUITS
 ON RADAR, GUIDANCE AND AVIONICS SYSTEMS. AVIONICS
 PANEL 49TH SYMPOSIUM (AGARD-CP-380)
 Lieu Conference : LISBON, PORTUGAL
 Date conference : 20-25 MAY 1985
 Auteurs : BERARDI L.; MERLANDO M.
 Affiliation : GRUPPO SISTEMI AVIONICI ED EQUIPAGGIAMENTI,
 AERITALIA, TORINO, ITALY (02);
 Type Document : Conference Paper
 Editeur : AGARD, NEUILLY-SUR-SEINE, FRANCE
 Source : NP. XVI+382; PP. 36/1-12; REF. 1; SPO. NATO; DP OCT.
 1985
 Resume : High performance integrated circuit technologies
 allow for dramatic improvement in speed and
 reduction in size of the integrated circuits. This
 fact results in the possibility to pack in a denser
 way the computing and decision-making functions of
 the avionics systems. Nevertheless it is still
 necessary to connect together the various system
 components, spread through the airframe, by means of
 an interfunction data transmission system. For these
 reasons the application of the new technologies to a
 mil-std-1553b data transmission system is
 considered. In particular the data management task
 allocated to the bus controller is described in four
 increasing levels of complexity, ranging from the
 minimum requirement to an 'expert' function
 including an high degree of configurability. The
 performance obtainable by implementing the functions
 in the current or new technologies, and with two
 different architectural solutions, are measured or
 estimated. The comparison among the obtained results
 shows that the new technologies not only improve the
 performances of the data transmission system but
 also allow to include an higher degree of
 intelligence in the function, extending in this way
 the application of mil-std-1553b to future advanced
 avionic systems.

-19- 597847 C.INSPEC

Titre Anglais : THE IMPACT OF VERY HIGH PERFORMANCE INTEGRATED
 CIRCUITS ON AVIONICS SYSTEM READINESS.
 Titre Conference: IMPACT OF VERY HIGH PERFORMANCE INTEGRATED CIRCUITS
 ON RADAR, GUIDANCE AND AVIONICS SYSTEMS. AVIONICS
 PANEL 49TH SYMPOSIUM (AGARD-CP-380)
 Lieu Conference : LISBON, PORTUGAL
 Date conference : 20-25 MAY 1985
 Auteurs : STRULL G.
 Affiliation : DIV. OF ADV. TECHNOL., WESTINGHOUSE ELECTR. CORP.,
 BALTIMORE, MD, USA

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Type Document : Conference Paper
 Editeur : AGARD, NEUILLY-SUR-SEINE, FRANCE
 Source : NP. XVI+382; PP. 35/1-4; SPD. NATO; DP OCT. 1985
 Resume : Vhpic represents a new systems/technology culture. With a philosophy of top-down design and bottom-up build, a vehicle is provided to avoid rapid device obsolescence so prevalent in the fast moving integrated circuit industry. However, to successfully and effectively design advanced systems in this manner, a design methodology is required that adequately addresses the challenge shown. Since everything from chip definition through application analysis is interactive with everything else, the challenge is to adequately keep track of all the perimeters and their relationship. The methodology by which design and analysis are accomplished is shown. The starting point is the systems architecture and its application software. From the architecture and application software the partitioning of the system into appropriate modules can be derived. From this an idea of the integrated circuits needed can be determined. This is the way the design proceeded on 'day one' of this technology revolution. However, at this point in time the author has a library of minicells and logic for creating macros so that he is well along in the chip development area. Therefore, he has the basis for simulation and the design continues in an interactive manner.

-20- 559521 C. INSPEC

Titre Anglais : AVIONICS, SOFTWARE, AND AUTOMATION.
 Titre Conference : IEEE EASCON'85: 18TH ANNUAL ELECTRONICS AND AEROSPACE SYSTEMS CONFERENCE (CAT. NO.85CH2213-7)
 Lieu Conference : WASHINGTON, DC, USA
 Date conference : 28-30 OCT. 1985
 Auteurs : HARRISON J. V. A.; LALA J. H.; BROCK L. D.
 Affiliation : CHARLES STARK DRAPER LAB. INC., CAMBRIDGE, MA, USA (03);

Type Document : Conference Paper
 Editeur : IEEE, NEW YORK, USA
 Source : NP. 378; PP. 93-109; SPD. IEEE; CCCC=CH2213-7/85/0000-0093\$01.00; DP 1985
 Resume : Summary form only given. The avionics functions in a launch vehicle or an orbital transfer vehicle are interpreted in a very broad context. They include all applications of electronics technology to signal, data, and symbolic processing, all of the information processing functions such as guidance, navigation, and control, propulsion control, decision support functions such as information fusion and planning, and displays, controls, and other man-machine interfaces. In the area of software, numerous opportunities exist to upgrade the present state of the art. Software complexity can be reduced by the development of very-high-level languages and other supporting tools such as automatic code generators. Software reliability can be improved through the development of fault-tolerant technologies. Automation represents a significant new opportunity to enhance system performance while, at the same time, containing system life-cycle costs.

-21- 515953 C. INSPEC

Titre Anglais : TOWARD AN AVIONICS SUPERSYSTEM.
 Type Document : Journal Article
 Titre Journal : DEF. ELECTRON. (USA)
 Source : VOL. 17, NO. 12; PP. 73-85; DP DEC. 1985
 Coden : DEELDH
 Issn : 0194-7885
 Resume : The long-term goal is to create a single, modular avionics system for communication, navigation, targeting, fire control, offensive and defensive ew, monitoring of systems status, and other functions.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

all controlled by a central computer and connected by a multiplex bus. One of the leading efforts in this field is the air force's pave pillar program, which is intended to create the architecture for such a system; its results will be integrated into army and navy avionics programs. Pave pillar has not yet reached the hardware stage; it has, however, produced some definite concepts for a fully integrated avionics system.

-22- 483169 C.INSPEC

Titre Anglais : FAULT TOLERANT HARDWARE/SOFTWARE ARCHITECTURE FOR FLIGHT CRITICAL FUNCTION (AGARD-LS-143).

Titre Conference: FAULT TOLERANT HARDWARE/SOFTWARE ARCHITECTURE FOR FLIGHT CRITICAL FUNCTION (AGARD-LS-143)

Lieu Conference : EDWARDS, USA

Date conference : 1-2 OCT. 1985

Type Document : Conference Proceedings

Editeur : AGARD, NEUILLY-SUR-SEINE, FRANCE

Source : NP. IV+146; SPO. AGARD; DP 1985

Resume : The following topics were dealt with: digital fly-by-wire experience; redundancy management of synchronous and asynchronous systems; software fault tolerance experiments; dependable avionic data transmission; multicomputer fault-tolerant systems using ada; design issues in data synchronous systems; digital fault-tolerant flight actuation systems; and design validation of fly-by-wire flight control systems. Abstracts of individual papers can be found under the relevant classification codes in this or other issues.

-23- 461976 C.INSPEC

Titre Anglais : GENERAL DYNAMICS CONVAIR DIVISION TOTALLY RECONFIGURABLE EMBEDDED COMPUTER.

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : MARKERT L.; HEDTKE P.; KUSEK J.

Affiliation : GEN. DYNAMICS CORP., SAN DIEGO, CA, USA (03);

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXXV+666; PP. 646-52; REF. 4; SPO. IEEE; AIAA; DP 1985

Resume : Modern avionics platforms require data processing systems that are extensible, maintainable, and affordable. A general-purpose embedded computer system is being developed for cruise missile and launch vehicle avionics applications. Major development objectives include implementation of us dept. Of defense embedded computer standards, a functionally flexible and modular design, and a producible system that is maintainable, reliable, and cost-effective. The hardware elements are described along with the details of their functional interfaces and the overall system architecture. An integration and maintenance support system is described. Software support and the implementation philosophy are also discussed.

-24- 461975 C.INSPEC

Titre Anglais : MODULAR AVIONICS PACKAGING STANDARDIZATION.

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : AUSTIN M.; MCNICHOLS J. K.

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXXV+666; PP. 641-5; SPO. IEEE; AIAA; DP 1985

Resume : Advanced integration technologies such as high speed data busing, built-in-test (bit), very large scale integration (vlsi) and very high speed integrated circuits (vhsic) provide an opportunity to further

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

integrate military avionics systems. It is pointed out, however, that these new technologies can be severely limited if placed in conventional avionics system architectures. To fully utilize the benefits of these technologies, modular packaging, standardization across multiple functions, multiple systems, and multiple airframes is needed. It is stressed that new methods of packaging this new generation of avionics are necessary to capitalize on benefits such as common sources, improved maintainability, increased reliability, and reduced life cycle costs.

-25- 461963 C.INSPEC

Titre Anglais : VLSI CHIP SET FOR HIGH-PERFORMANCE AVIONIC COMPUTERS.

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : FORDE S. J.; HILMANTEL M. A.

Affiliation : SANDERS ASSOCIATES INC., NASHUA, NH, USA (02);

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXV+666; PP. 569-72; REF. 1; SPO. IEEE; AIAA; DP 1985

Resume : Advanced CMOS processing with line widths of two micrometers makes possible a VLSI implementation of avionics computers. Using a chip set designed with 6000-gate gate arrays has led to the design of a very high-performance, yet small, low-power, cost-effective, and highly flexible avionics computer. The chip set comprises four chips: the microsequencer, the arithmetic and logic unit, the operand generator unit, and the memory controller unit. The four chips can be configured to implement the US Air Force standard instruction set architecture (MIL-STD-1750A, notice 1) and emulate the US Navy standard computers (AN/Uyk-20, AN/Uyk-14, and AN/Uyk-44). An overview of the chip architectures is presented.

-26- 461938 C.INSPEC

Titre Anglais : INTEGRATED AVIONICS FOR ADVANCED ARMY ROTORCRAFT.

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : EVERSOLE W. L.; KICZUK W. F.; LAMBRECHT J. A.;

RIVARD R. L.; WILLIAMS J. J.;

Affiliation : TEXAS INSTRUM. INC., DALLAS, TX, USA (06);

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXV+666; PP. 352-8; SPO. IEEE; AIAA; DP 1985

Resume : An approach to the development of an advanced avionics architecture to meet the functional requirements of the US Army's next generation of rotorcraft (LHX) is presented. Mission requirements are briefly discussed to identify the functional partitioning of the LHX system. This is followed by a brief overview of each subsystem. The requirements of the processing algorithms are discussed to quantitatively identify the performance requirements of the avionics architecture. Also described is a multiprocessor, multisensor architecture involving high bandwidth sensors, low bandwidth sensors, and a hierarchy of processing structures and interconnections that provides the flexibility, reliability, availability, and fault tolerance within the power, volume, and weight constraints imposed by LHX.

-27- 461925 C.INSPEC

Titre Anglais : TEST EXPERIENCE ON AN ULTRARELIABLE COMPUTER COMMUNICATION NETWORK FOR FLIGHT CONTROL SYSTEM

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

APPLICATION!

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : ABBOTT L. W.

Affiliation : AMES RES. CENTER, NASA, EDWARDS, CA, USA

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXXV+666; PP. 233-8; REF. 6; SPD. IEEE; AIAA; DP 1985

Resume : The dispersed sensor processing mesh (dspm) is an experimental, ultrareliable, fault-tolerant computer communications network that exhibits an organic-like ability to regenerate itself after suffering damage. The regeneration is accomplished by two routines-grow and repair. The author discusses the dspm concept for achieving fault tolerance and provides a brief description of the mechanization of both the experiment and the six-node experimental network. The main topic is the system performance of the growth algorithm contained in the growth routing. The characteristics imbued to dspm by the growth algorithm are also discussed. Data from an experimental dspm network and software simulation of larger dspm-type networks are used to examine the inherent limitation on growth time by the growth algorithm and the relationship of growth time to network size and topology.

-28- 45118 C.INSPEC

Titre Anglais : USING ADA FOR A DISTRIBUTED, FAULT TOLERANT SYSTEM #AVIONICS APPLICATIONS!

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : DEWOLF J. B.; SODANO N. M.; WHITTREDGE R. S.

Affiliation : CHARLES STARK DRAPER LAB., INC., CAMBRIDGE, MA, USA (03);

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXXV+666; PP. 477-84; REF. 6; SPD. IEEE; AIAA; DP 1985

Resume : The authors present a snapshot of one project's concerns and proposed solutions for using ada to develop system software for a distributed damage- and fault-tolerant processing architecture. The project is the nasa-sponsored advanced information processing system (aips). The authors also describe how the aips system software can be written in ada using a standard uniprocessor runtime support package. The system software provides certain commonly used services to the applications programmer beyond those inherent in the ada language definition. In addition, it provides network transparency for interfunction communication. Implementing these services in ada raises certain issues relating to the runtime support package. The current strategy for responding to these issues is described and recommendations are made for improving ada runtime support features.

-29- 451107 C.INSPEC

Titre Anglais : ULTRARELIABLE FAULT-TOLERANT CONTROL SYSTEMS #AEROSPACE CONTROL!

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : WEBSTER L. D.; SLYKHOUSE R. A.; BOOTH L. A.; CARSON T. W.; DAVIS G. J.

Affiliation : AMES RES. CENTER, NASA, MOFFETT FIELD, CA, USA (06);

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Source : NP. XXXV+666; PP. 239-46; REF. 9; SPO. IEEE; AIAA; DP 1985

Resume : An ultrareliable fault-tolerant control system (ufics) concept is described using a systems design philosophy which allows the development of system structures containing virtually no common elements. Common elements limit achievable system reliability and can cause catastrophic loss of fault-tolerant system function. The ufics concept provides the means for removing common system elements by permitting the elements of the system to operate as independent, uncoupled entities, multiple versions of the application program are run on dissimilar hardware. Fault tolerance is achieved through the use of static redundancy management.

-30- 451102 C.INSPEC

Titre Anglais : DEVELOPMENT TOOLS: CASE STUDY FOR LARGE SYSTEMS.

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : HORNBACK K.

Affiliation : INSTRUM. DIV., LEAR SIEGLER, GRAND RAPIDS, MI, USA

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXXV+666; PP. 167-74; REF. 8; SPO. IEEE; AIAA; DP 1985

Resume : Software development tools can be an important aid in controlling the complexity of large digital avionics systems. The author describes the successful application of modern software tools to the development of the flight management computer system for the boeing 737-300 aircraft. Tools were used to increase productivity and quality during the entire software life cycle. Source code management tools provided thorough, ongoing configuration management of code. Static analysis and path coverage of the source aided in meeting stringent verification requirements. Fourth-generation language techniques were used to produce many of the tools cost-effectively; and text formatting tools were used to increase documentation productivity.

-31- 451101 C.INSPEC

Titre Anglais : SOFTWARE TRACEABILITY, REQUIREMENTS TESTABILITY, AND AUDITING MODEL.

Titre Conference: PROCEEDINGS OF THE AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE (CAT. NO. 84CH-2132-9)

Lieu Conference : BALTIMORE, MD, USA

Date conference : 3-6 DEC. 1984

Auteurs : SCIORTINO J.; DUNNING D.

Affiliation : ARINC RES. CORP., ANNAPOLIS, MD, USA (02);

Type Document : Conference Paper

Editeur : AIAA, NEW YORK, USA

Source : NP. XXXV+666; PP. 159-66; REF. 3; SPO. IEEE; AIAA; DP 1985

Resume : Software development for both management information systems (mis) and embedded computer systems (ecs) has suffered dramatic cost and time overruns, has resulted in user dissatisfaction, and has often yielded software that does not work in spite of extra time and money spent. Arinc research has developed a software traceability tool for reducing risks in software development. This software traceability, requirements testability, and auditing (strata) model is a microcomputer-based tracking system developed to support and automate the top-down structured approach to software development. A brief description is given of the environment of embedded systems development and of the advantages of applying a tool like strata to the software development of digital avionics system.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

-32- 442868 C.INSPEC

Titre Anglais : DIGITAL AVIONICS FOR MODERN AIRCRAFT-A CASE STUDY
INTO THE PROBLEMS AND PROMISE OF AIRCRAFT
ELECTRONICS.
Titre Conference: PROCEEDINGS OF THE IEEE 1985 NATIONAL AEROSPACE AND
ELECTRONICS CONFERENCE NAECON 1985 (CAT. NO.
85CH2189-9)
Lieu Conference : DAYTON, OH, USA
Date conference : 20-24 MAY 1985
Auteurs : ARCHER H. S.
Affiliation : LOCKHEED GEORGIA CO., MARIETTA, GA, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 2 VOL. 1595; PP. 144-51 VOL.1; REF. 1; SPD.
IEEE; AEROSP. ELECTRON. SYST. SOC; CCCC=
0547-3578/85/0000-0144\$01.00; DP 1985

Resume : With decreasing static stability margins and
increasingly rigorous mission requirements, the role
of aircraft electronics has become vital. The author
gives a brief historical perspective of the
evolution of avionics and notes the harsh
environmental constraints under which current
avionics must operate. The case of spatially
redundant integrated racks composed of standardized
modules is made. A state-of-the-art design approach
is then presented, followed by a discussion of an
advanced system that will be typical of aircraft in
the 1990s. The topics discussed include: designing
for thermal management, distributed processing,
built-in-test requirements, and redundant systems
for flight-critical applications.

-33- 442861 C.INSPEC

Titre Anglais : DISTRIBUTED MEMORY NETWORK: AN 8 GIGABIT FIBER OPTIC
TIGHTLY COUPLED SYSTEM.
Titre Conference: PROCEEDINGS OF THE IEEE 1985 NATIONAL AEROSPACE AND
ELECTRONICS CONFERENCE NAECON 1985 (CAT. NO.
85CH2189-9)
Lieu Conference : DAYTON, OH, USA
Date conference : 20-24 MAY 1985
Auteurs : FOLMAR R. J.
Affiliation : HARRIS CORP., MELBOURNE, FL, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 2 VOL. 1595; PP. 91-4 VOL.1; SPD. IEEE; AEROSP.
ELECTRON. SYST. SOC; CCCC=
0547-3578/85/0000-0091\$01.00; DP 1985

Resume : The distributed memory network (dmn) described is an
8-gb/s fiber optic, memory connected local area
network. It is designed for avionic applications and
vhic signal processor speeds. System operation,
memory connected communication, fiber optic data
link and fault-tolerant features are described. An
engineering development model of this network that
is currently in the fabrication phase is described.

-34- 433231 C.INSPEC

Titre Anglais : SOFTWARE TOOLS-A WAY TO CONTROL COMPLEXITY ON LARGE
SOFTWARE PROJECTS.
Titre Conference: PROCEEDINGS OF THE IEEE 1985 NATIONAL AEROSPACE AND
ELECTRONICS CONFERENCE NAECON 1985 (CAT. NO.
85CH2189-9)
Lieu Conference : DAYTON, OH, USA
Date conference : 20-24 MAY 1985
Auteurs : HORNBAUGH K.
Affiliation : LEAR SIEGLER, GRAND RAPIDS, MI, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 2 VOL. 1595; PP. 813-20 VOL.1; REF. 12; SPD.
IEEE; AEROSP. ELECTRON. SYST. SOC; CCCC=
0547-3578/85/0000-0813\$01.00; DP 1985

Resume : The size of avionics software development projects
has grown dramatically over the last several years.
The complexity of these projects has increased even
faster. The author analyzes how software development

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

tools can be used to control the complexity and improve productivity during the development of large avionics systems. Major tool systems examined include source code configuration management, problem report database, static analysis of source code, data dictionaries, regression test managers, methodology automation, and electronic mail. Online databases are produced as a side effect of many of these tools. When enough databases are present, a 'project knowledge base' emerges, which can be used and extended to provide additional management and tracking information. These computer-based tool can be used to accurately track the tens of thousands of details related to the project, and to provide timely progress feedback to management. Examples from an actual project are described.

-35- 433224 C.INSPEC

Titre Anglais : A NEW APPROACH TO ENSURING DETERMINISTIC PROCESSING IN AN INTEGRATED AVIONICS SOFTWARE SYSTEM.
Titre Conference: PROCEEDINGS OF THE IEEE 1985 NATIONAL AEROSPACE AND ELECTRONICS CONFERENCE NAECON 1985 (CAT. NO. 85CH2189-9)

Lieu Conference : DAYTON, OH, USA
Date conference : 20-24 MAY 1985
Auteurs : ELLIS J. R.
Affiliation : HARRIS CORP., MELBOURNE, FL, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 2 VOL. 1595; PP. 756-63 VOL. 1; REF. 4; SPO. IEEE; AEROSP. ELECTRON. SYST. SOC: CCCC-0547-3578/85/0000-0756\$01.00; DP 1985

Resume : The traditional approach to avionics software systems has been to ensure identical time for each cycle through the code regardless of the path taken. This cyclic executive approach is not compatible with future avionics software with continually increasing functional integration. The author describes a software architecture used in an integrated avionics software system for the agusta a-129 light attack helicopter. The architecture is presented as the executive functions required for integrated deterministic and statistical processing in a single computer; it supports integrated fault detection for software and/or hardware failures. The architecture supports use of high-order programming languages, potentially shorter system development times through the introduction of greater development concurrency, and greater flexibility and lower costs in software maintenance, without sacrificing operational reliability.

-36- 359807 C.INSPEC

Titre Anglais : FAULT TOLERANT COMPUTER SYSTEM FOR THE A129 HELICOPTER.

Auteurs : JOHNSON B. W.; JULICH P. M.
Affiliation : DEPT. OF ELECTR. ENG., VIRGINIA UNIV., CHARLOTTESVILLE, VA, USA
Type Document : Journal Article
Titre journal : IEEE TRANS. AEROSP. AND ELECTRON. SYST. (USA)
Source : VOL. AES-21, NO. 2; PP. 220-9; REF. 5; CCCC-0018-9251/85/0300-0220\$01.00; DP MARCH 1985

Coden : IEARAX
Issn : 0018-9251

Resume : A description of the design and an analysis of the fault-tolerance characteristics of the a129 integrated multiplex system (ims) are presented. The a129 ims is a computer system designed to implement automatic flight control, navigation, system monitoring, and other flight-critical and mission-related tasks. The fault-tolerance design philosophy has been to implement the majority of the fault detection and redundancy management features in software as opposed to hardware. The resulting system has been shown, via a markov reliability

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

analysis, to possess high reliability during a
three-hour mission.

-37- 321795 C.INSPEC

Titre Anglais : ACHIEVING GREATER PRODUCTIVITY THROUGH SOFTWARE
TOOLS.
Titre Conference: PROCEEDINGS OF THE DIGITAL EQUIPMENT COMPUTER USERS
SOCIETY EUROPE 1984
Lieu Conference : AMSTERDAM, NETHERLANDS
Date conference : 24-28 SEPT. 1984
Auteurs : HORNBAUGH K.
Affiliation : LEAR SIEGLER INSTRUM. DIV., GRAND RAPIDS, MI, USA
Type Document : Conference Paper
Editeur : DECUS EUROPE, PETIT-LANCY, SWITZERLAND
Source : NP. VI+577; PP. 374-85; REF. 6; DP 1983
Resume : While papers abound on the theoretical and prototype
usage of software development tools, there have been
very few case studies showing the integrated use of
commercially available tools on large real-world
projects. A new set of problems arise when tools and
techniques developed for small projects are used on
projects an order of magnitude larger. This paper
describes the successful application of modern tools
and methodologies to a very large, real time
avionics system-what worked, what didn't, and what
one would change if one could do it over again. The
biggest paybacks sometimes came from unexpected
areas; as did many of the problems. Dec's vms
software development tools and vax information
architecture provided much of the foundation.
Configuration management, automation of structured
analysis and design, static analysis of source, and
automated documentation are some examples of tool
usage examined.

-38- 307036 C.INSPEC

Titre Anglais : DOCUMENTATION AND SEPARATE TEST PROGRAM DEVELOPMENT
IS MOST IMPORTANT FOR TEST/MAINTENANCE OF FLIGHT
SOFTWARE.
Titre Conference: AGARD CONFERENCE PROCEEDINGS NO. 361. DESIGN FOR
TACTICAL AVIONICS MAINTAINABILITY (AGARD-CP-361)
Lieu Conference : BRUSSELS, BELGIUM
Date conference : 7-10 MAY 1984
Auteurs : GUSMANN B.; SANDNER N.
Affiliation : LITEF, FREIBURG, GERMANY (O2);
Type Document : Conference Paper
Editeur : AGARD, NEUILLY-SUR-SEINE, FRANCE
Source : NP. XI+294; PP. 17/1-10; DP OCT. 1984
Resume : At litef, operational flight software is in either
the maintenance or development phase for various
systems for transport aircraft and military
applications. Support software is in the development
and maintenance phase for the new tornado main
computer. Most important for test and maintenance
phases are well defined development phases with
standardized documentations supported by computer
based tools. The litef software development is based
on a software handbook consisting of software
guidelines, methods/techniques and automated tools.
The authors focus on the documentation throughout
the lifecycle and the importance of independent
testing.

-39- 269712 C.INSPEC

Titre Anglais : SOFTWARE MAINTENANCE WORKSHOP RECORD (CAT. NO.
83CH1982-8).
Titre Conference: SOFTWARE MAINTENANCE WORKSHOP RECORD (CAT. NO.
83CH1982-8)
Lieu Conference : MONTEREY, CA, USA
Date conference : 6-8 DEC. 1983
Auteurs : ARNOLD R. S.
Type Document : Conference Proceedings
Editeur : IEEE COMPUT. SOC. PRESS, SILVER SPRING, MD, USA
Source : NP. XV+302; SPO. IEEE; NBS; NAVAL POSTGRADUATE
SCHOOL; ACM; DP 1984

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY
INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Isbn : 0-8186-0510-3
Resume : The following topics were dealt with: approaches for providing maintainable software; tools for software maintenance; program evolution; software maintenance and testing practices; software and maintenance management; understanding and documenting software; maintenance in a database environment; software maintenance in Sweden; software maintenance in Japan; and avionics software maintenance case studies. 51 papers were presented, of which 45 are published in full in the present proceedings, and 6 as abstracts only. Twelve papers that were not presented are also included in these proceedings. Abstracts of individual papers can be found under the relevant classification codes in this or other issues.

-40- 185722 C.INSPEC
Titre Anglais : THE SPACE SHUTTLE PRIMARY COMPUTER SYSTEM.
Auteurs : SPECTOR A.; GIFFORD D.
Affiliation : DEPT. OF COMPUTER SCI., CARNEGIE-MELLON UNIV., PITTSBURGH, PA, USA
Type Document : Journal Article
Titre journal : COMMUN. ACM (USA)
Source : VOL.27, NO.9; PP. 872-900; REF. 5; CCCC=0001-0782/84/0900-0874/75C; DP SEPT. 1984
Codex : CACMA2
Issn : 0001-0782
Resume : IBM's federal systems division is responsible for supplying 'error-free' software for NASA's space shuttle program. The shuttle's primary avionics software system (pass) is the highly fault-tolerant, on-board software that controls most aspects of shuttle operation. The on-board system (called the dps, for data processing system) utilizes five computers, which are known as gpcs (general-purpose computers). The pass resides in a maximum of four of these at any one time. Since the software needed to support an entire mission would be too large to occupy the primary memory, it has been divided into eight overlays that make up the operational programs of pass. The software has an operating system written in assembler and applications written in hal/s, a high-order language developed by Intermetrics inc. Of Cambridge, Massachusetts.

-41- 153082 C.INSPEC
Titre Anglais : A HIGH-SPEED MIL-STD-1750A CMOS/SOS MICROPROCESSOR #AVIONICS APPLICATION!
Titre Conference: PROCEEDINGS OF THE IEEE/AIAA 5TH DIGITAL AVIONICS SYSTEMS CONFERENCE
Lieu Conference : SEATTLE, WA, USA
Date conference : 31 OCT.-3 NOV. 1983
Auteurs : RASSET T. L.; HANNA W. A.; WALLACE L. N.
Affiliation : MCDONNELL DOUGLAS ASTRONAUTICS CO., HUNTINGTON BEACH, CA, USA
Type Document : Conference Paper
Editeur : IEEE, NEW YORK, USA
Source : NP. 826; PP. 18.6/1-7; REF. 3; SPD. IEEE; CCCC=CH1839-0/83/0000-0085\$01.00; DP 1983
Resume : A high-speed, low-power, hardened mil-std-1750a (usaf) cmos/sos chip set is discussed. The chip set contains the following options: timers a and b; trigger-go counter; startup row control; and dma control. Special elements have been built into the chip set to aid testing and hardware/software integration. Both a hardened and a space-hardened version of the chip set have been developed. To support the chip set, an integrated set of hardware and software development tools, including a processor monitor system, has been developed.

-42- 135027 C.INSPEC
Titre Anglais : PROCEEDINGS OF THE IEEE/AIAA 5TH DIGITAL AVIONICS SYSTEMS CONFERENCE.

COMPUTING SYSTEM CONFIGURATIONS FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS

Titre Conference: PROCEEDINGS OF THE IEEE/AIAA 5TH DIGITAL AVIONICS
 SYSTEMS CONFERENCE
 Lieu Conference : SEATTLE, WA, USA
 Date conference : 31 OCT.-3 NOV. 1983
 Type Document : Conference Proceedings
 Editeur : IEEE, NEW YORK, USA
 Source : NP. 826; SP0. IEEE; DP 1983
 Resume : The following topics are dealt with: advanced
 avionics systems; avionics for v/stol and
 rotorcraft; software management; simulation; digital
 flight control systems; flight software; integrated
 communications, navigation and identification
 systems; sensors and signal processing; data buses;
 on-board monitoring and support; fault-tolerant
 avionics; digital systems; integrated crew stations;
 traffic alert and collision avoidance systems;
 commercial aircraft systems; and vhsic applications.
 142 papers were presented, of which 114 are
 published in full in the present proceedings, and 4
 as abstracts only. Abstracts of individual papers
 can be found under the relevant classification codes
 in this or other issues.

-43- 33262 C.INSPEC

Titre Anglais : TECHNICAL EVALUATION REPORT ON THE GUIDANCE AND
 CONTROL PANEL 35TH SYMPOSIUM ON ADVANCES IN GUIDANCE
 AND CONTROL SYSTEMS.
 Auteurs : REDIESS H. A.
 Organisme auteur : AGARD, NEUILLY SUR SEINE, FRANCE
 Type Document : Report
 Numeros : RPT AGARD-AR-195
 Source : NP. IV+11; DP JULY 1983
 Resume : Many significant advances in optimal control theory,
 synthesis techniques and design methodology have
 taken place since the last symposium held in this
 technical area in 1973. The rapidly developing
 technologies in computation, data distribution,
 computer aided design methods and data basis now
 permit application of theories and synthesis methods
 heretofore impractical. The increased emphasis on
 functional and performance capability at reduced
 cost suggests application of technologies and
 methods for more common use of information and
 higher levels of integration. The purpose of the
 meeting was to review and discuss all aspects of
 those emerging technologies ranging from theory
 through applications including aircraft, space
 vehicles, and unmanned vehicles.

| REPORT DOCUMENTATION PAGE | | | | | | | | | |
|-------------------------------|--|----------------------|--|------------------|-------------------|--------------------|----------|---------------------------|--------|
| 1. Recipient's Reference | 2. Originator's Reference | 3. Further Reference | 4. Security Classification of Document | | | | | | |
| | AGARD-LS-158 | ISBN 92-835-0464-X | UNCLASSIFIED | | | | | | |
| 5. Originator | Advisory Group for Aerospace Research and Development North Atlantic Treaty Organization 7 rue Ancelle, 92200 Neuilly sur Seine, France | | | | | | | | |
| 6. Title | COMPUTING SYSTEMS CONFIGURATION FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS | | | | | | | | |
| 7. Presented at | | | | | | | | | |
| 8. Author(s)/Editor(s) | Various | | 9. Date | | | | | | |
| | | | June 1988 | | | | | | |
| 10. Author's/Editor's Address | Various | | 11. Pages | | | | | | |
| | | | 166 | | | | | | |
| 12. Distribution Statement | This document is distributed in accordance with AGARD policies and regulations, which are outlined on the Outside Back Covers of all AGARD publications. | | | | | | | | |
| 13. Keywords/Descriptors | <table border="0"> <tr> <td>Missile guidance</td> <td>Control equipment</td> </tr> <tr> <td>Guidance computers</td> <td>Avionics</td> </tr> <tr> <td>Computer systems programs</td> <td>Design</td> </tr> </table> | | | Missile guidance | Control equipment | Guidance computers | Avionics | Computer systems programs | Design |
| Missile guidance | Control equipment | | | | | | | | |
| Guidance computers | Avionics | | | | | | | | |
| Computer systems programs | Design | | | | | | | | |
| 14. Abstract | <p>Modern military air vehicles have to comply with sophisticated performance requirements. As a result, full advantage must be taken of the rapid advances in computer hardware/software and future micro-electronics technologies.</p> <p>New design and development strategies must be implemented in order to obtain the overall performance benefits offered by advanced integrated systems for guidance and control, avionics, weapon delivery and tactical performance management.</p> <p>In a two-day programme this Lecture Series will address some issues which have demonstrated notable and outstanding advances in the field of computing system design, design tools and techniques, computers, data buses, and architectures. In particular, the second day's programme will show how technological advances have enabled the design of a modern computing system architecture. Future trends and new directions will be subjects for round table discussions.</p> <p>This Lecture Series, sponsored by the Guidance and Control Panel of AGARD, has been implemented by the Consultant and Exchange Programme of AGARD.</p> | | | | | | | | |

| | | | |
|--|--|--|--|
| <p>AGARD Lecture Series No.158 Advisory Group for Aerospace Research and Development, NATO COMPUTING SYSTEMS CONFIGURATION FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS Published June 1988 166 pages</p> <p>Modern military air vehicles have to comply with sophisticated performance requirements. As a result, full advantage must be taken of the rapid advances in computer hardware/software and future micro-electronics technologies.</p> <p>New design and development strategies must be implemented in order to obtain the overall performance P.T.O</p> | <p>AGARD-LS-158</p> <p>Missile guidance Guidance computers Computer systems programs Control equipment Avionics Design</p> | <p>AGARD Lecture Series No.158 Advisory Group for Aerospace Research and Development, NATO COMPUTING SYSTEMS CONFIGURATION FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS Published June 1988 166 pages</p> <p>Modern military air vehicles have to comply with sophisticated performance requirements. As a result, full advantage must be taken of the rapid advances in computer hardware/software and future micro-electronics technologies.</p> <p>New design and development strategies must be implemented in order to obtain the overall performance P.T.O</p> | <p>AGARD-LS-158</p> <p>Missile guidance Guidance computers Computer systems programs Control equipment Avionics Design</p> |
| <p>AGARD Lecture Series No.158 Advisory Group for Aerospace Research and Development, NATO COMPUTING SYSTEMS CONFIGURATION FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS Published June 1988 166 pages</p> <p>Modern military air vehicles have to comply with sophisticated performance requirements. As a result, full advantage must be taken of the rapid advances in computer hardware/software and future micro-electronics technologies.</p> <p>New design and development strategies must be implemented in order to obtain the overall performance P.T.O</p> | <p>AGARD-LS-158</p> <p>Missile guidance Guidance computers Computer systems programs Control equipment Avionics Design</p> | <p>AGARD Lecture Series No.158 Advisory Group for Aerospace Research and Development, NATO COMPUTING SYSTEMS CONFIGURATION FOR HIGHLY INTEGRATED GUIDANCE AND CONTROL SYSTEMS Published June 1988 166 pages</p> <p>Modern military air vehicles have to comply with sophisticated performance requirements. As a result, full advantage must be taken of the rapid advances in computer hardware/software and future micro-electronics technologies.</p> <p>New design and development strategies must be implemented in order to obtain the overall performance P.T.O</p> | <p>AGARD-LS-158</p> <p>Missile guidance Guidance computers Computer systems programs Control equipment Avionics Design</p> |

| | |
|--|--|
| <p>benefits offered by advanced integrated systems for guidance and control, avionics, weapon delivery and tactical performance management.</p> <p>In a two-day programme this Lecture Series will address some issues which have demonstrated notable and outstanding advances in the field of computing system design, design tools and techniques, computers, data buses, and architectures. In particular, the second day's programme will show how technological advances have enabled the design of a modern computing system architecture. Future trends and new directions will be subjects for round table discussions.</p> <p>This Lecture Series, sponsored by the Guidance and Control Panel of AGARD, has been implemented by the Consultant and Exchange Programme of AGARD.</p> <p>ISBN 92-835-0464-X</p> | <p>benefits offered by advanced integrated systems for guidance and control, avionics, weapon delivery and tactical performance management.</p> <p>In a two-day programme this Lecture Series will address some issues which have demonstrated notable and outstanding advances in the field of computing system design, design tools and techniques, computers, data buses, and architectures. In particular, the second day's programme will show how technological advances have enabled the design of a modern computing system architecture. Future trends and new directions will be subjects for round table discussions.</p> <p>This Lecture Series, sponsored by the Guidance and Control Panel of AGARD, has been implemented by the Consultant and Exchange Programme of AGARD.</p> <p>ISBN 92-835-0464-X</p> |
| <p>benefits offered by advanced integrated systems for guidance and control, avionics, weapon delivery and tactical performance management.</p> <p>In a two-day programme this Lecture Series will address some issues which have demonstrated notable and outstanding advances in the field of computing system design, design tools and techniques, computers, data buses, and architectures. In particular, the second day's programme will show how technological advances have enabled the design of a modern computing system architecture. Future trends and new directions will be subjects for round table discussions.</p> <p>This Lecture Series, sponsored by the Guidance and Control Panel of AGARD, has been implemented by the Consultant and Exchange Programme of AGARD.</p> <p>ISBN 92-835-0464-X</p> | <p>benefits offered by advanced integrated systems for guidance and control, avionics, weapon delivery and tactical performance management.</p> <p>In a two-day programme this Lecture Series will address some issues which have demonstrated notable and outstanding advances in the field of computing system design, design tools and techniques, computers, data buses, and architectures. In particular, the second day's programme will show how technological advances have enabled the design of a modern computing system architecture. Future trends and new directions will be subjects for round table discussions.</p> <p>This Lecture Series, sponsored by the Guidance and Control Panel of AGARD, has been implemented by the Consultant and Exchange Programme of AGARD.</p> <p>ISBN 92-835-0464-X</p> |